

Computing the Skorokhod Distance between Polygonal Traces

Rupak Majumdar Vinayak S. Prabhu
 MPI-SWS
 {rupak,vinayak}@mpi-sws.org

Abstract

The *Skorokhod distance* is a natural metric on traces of continuous and hybrid systems. For two traces, from $[0, T]$ to values in a metric space \mathcal{O} , it measures the best match between the traces when allowed continuous bijective timing distortions. Formally, it computes the infimum, over all timing distortions, of the maximum of two components: the first component quantifies the *timing discrepancy* of the timing distortion, and the second quantifies the mismatch (in the metric space \mathcal{O}) of the values under the timing distortion. Skorokhod distances appear in various fundamental hybrid systems analysis concerns: from definitions of hybrid systems semantics and notions of equivalence, to practical problems such as checking the closeness of models or the quality of simulations. Despite its popularity and extensive theoretical use, the *computation* problem for the Skorokhod distance between two finite sampled-time hybrid traces has remained open.

We address in this work the problem of computing the Skorokhod distance between two polygonal traces (these traces arise when sampled-time traces are completed by linear interpolation between sample points). We provide the first algorithm to compute the exact Skorokhod distance when trace values are in \mathbb{R}^n for the L_1 , L_2 , and L_∞ norms. Our algorithm, based on a reduction to Fréchet distances, is fully polynomial-time, and incorporates novel polynomial-time procedures for a set of geometric primitives in \mathbb{R}^n over the three norms.

1 Introduction

In approximation theories, we aim to quantify the difference between hybrid systems by defining a metric on traces. Given finite system traces $x : [0, T] \mapsto \mathcal{O}$ mapping the time interval $[0, T]$ to some metric space \mathcal{O} , one simple method to obtain a metric is to take the pointwise trace value difference: the difference between two traces x and y is then $\sup_t \mathcal{D}(x(t), y(t))$ where \mathcal{D} is the metric associated with the metric space \mathcal{O} . The restriction that we compare the value of x at time t to the value of y at the *same* time t often too restrictive: a trace can have a large distance from its infinitesimally time-shifted version. This motivates the study of the *Skorokhod* metric [Sko56], which allows for “wobble-room” in both the trace values *and* in the timeline. The distortion of the timeline is specified by a *retiming* function r which is a continuous bijective strictly increasing function from \mathbb{R}_+ to \mathbb{R}_+ . Using the retiming function, we obtain the *retimed trace* $x(r(t))$ from the original trace $x(t)$. Intuitively, in the retimed trace $x(r(t))$, we see exactly the same values as before, in exactly the same order, but the time duration between two values might now be different than the corresponding duration in the original trace. The amount of distortion for the retiming r is given by $\sup_{t \geq 0} |r(t) - t|$. Using retiming functions, the Skorokhod distance between two traces

x and y is defined to be the least value over all possible retimings r of:

$$\max \left(\sup_{t \in [0, T]} |r(t) - t|, \sup_{t \in [0, T]} \mathcal{D}(x(r(t)), y(t)) \right).$$

The Skorokhod distance thus incorporates two components: the first component quantifies the *timing discrepancy* of the timing distortion required to “match” two traces, and the second quantifies the *value mismatch* (in the metric space \mathcal{O}) of the values under the timing distortion.

The Skorokhod metric has been widely used in the semantics and analysis of continuous, boolean, stochastic, and hybrid systems [CB02, Bro98]. Despite its popularity, the *computation* of the Skorokhod metric has not been studied, except for some discrete-time cases (where there is a folklore dynamic programming algorithm); even the exact computability for continuous piecewise-linear traces (called *polygonal traces*), which arise when time-sampled traces are completed by linear interpolation, was open. We note that even for linear traces, the space of retiming functions is infinite; and that a linear trace, after retiming, need not remain linear.

Our Contributions. In this paper, we show a fully polynomial-time algorithm for the computation of the Skorokhod metric for polygonal traces for L_1 , L_2 , and L_∞ norms. Our algorithm reduces the computation of Skorokhod distances between continuous traces in a normed space to computing a related distance, called the *Fréchet distance*, between curves (in a different normed space). Fréchet distances have been studied extensively in computational geometry (see *e.g.* [BBW08, CdVE⁺10, MSSZ11]). A celebrated paper by Alt and Godau [AG95] gave a sketch of a polynomial-time algorithm for polygonal curves in \mathbb{R}^2 in the L_2 norm. We provide a generalization of this algorithm parameterized by a set of geometric primitives over the underlying metric space of curves. We also provide polynomial-time algorithms for these geometric primitives in \mathbb{R}^n for the L_1 , L_2 , and L_∞ metrics, as well as for two derived metrics L_1^S and L_2^S which are required for computing the Skorokhod distance. These algorithms involve techniques from linear programming (for L_1 , L_∞ and L_1^S norms), and from vector algebra and convex geometry (for the L_2 and L_2^S norms). Together, we get a fully polynomial time algorithm for the Skorokhod distance in \mathbb{R}^n for the L_1 , L_2 , and L_∞ metrics. In addition, our constructions also provide a fully polynomial time algorithm for computing the Fréchet distance between finite polygonal curves in \mathbb{R}^n for these metrics. For practical applications where only constant window retimings are of interest (*i.e.*, where the k -th affine segment of y can only be matched to the portion of x between the $(k - W)$ -th and $(k + W)$ -th affine segments, for W a constant), our algorithm for the Skorokhod distance runs in time $\mathcal{O}(m \cdot \log(m))$, for a constant dimensional space \mathbb{R}^n , where m is the number of affine line segments in the polygonal traces. The corresponding decision problem runs in linear time.

Our treatment in this paper is self-contained – we do not assume any background in computational geometry. We derive, and prove where necessary, missing results, and generalizations of the results, that were sketched in [AG95] for \mathbb{R}^2 and L_2 , which are needed in our algorithm for computing the Skorokhod distances in \mathbb{R}^n for the L_1 , L_2 , and L_∞ metrics. These steps require us to do a careful and detailed analysis of the the [AG95] algorithm sketch, filling in missing details, and correcting inaccuracies, in order to generalize to higher dimensions for the various norms.

Related Work. Metrics between traces are the basis for robustness and abstraction for hybrid systems [Bro98, CB02, Tab09]. More recently, distance metrics have been used to guide test generation [FSUY12] and for conformance testing between different models of a system [AMF14, AHF⁺14].

Metrics related to the Skorokhod metric have been studied in the context of timed systems [CP13, CIM14]; our results are orthogonal. *Dynamic Time Warping* [Mül07, BC96] is a discrete *sum* measure (it aggregate the discrepancies over the timeline), as opposed to the max-measure of Fréchet distances, between discrete time sequences and has been used heavily in signal processing and data mining. Sum measures take the sum of the trace differences after retiming, as opposed to considering the maximal (retimed) trace difference. The discrete time Dynamic Time Warping distance be computed using dynamic programming, and has approximation algorithms which are efficient [SC07]. The continuous analog of the Dynamic Time Warping sum measure between curves is explored, and an algorithm presented for polygonal curves, in [EVF07].

Outline of the Paper. In Sections 2 and 3 we introduce the Skorokhod and Fréchet metrics, and show how the computation of the Skorokhod metric between two continuous traces in a normed space can be reduced to the computation of the Fréchet metric between two corresponding curves in a related normed space. In Section 4 we present the general algorithms for polygonal curves in \mathbb{R}^n (for five different norms) for the Fréchet distance *decision problem*; and for the computation of the *value* of the Fréchet distance. The value computation algorithm is parameterized by a set of geometric primitives over the underlying metric space of curves. The computation procedures for these geometric primitives are obtained in Section 5 for the five norms. Section 6 ties everything together for the Skorokhod distance computation problem for the norms L_1, L_2 , and L_∞ . We conclude with a discussion of the paper in Section 7.

2 Skorokhod Distances between Traces

We begin by defining the Skorokhod distance between finite traces. A (finite) *trace* $x : [T_i, T_e] \rightarrow \mathcal{O}$ is a mapping from a finite time interval $[T_i, T_e]$, with $0 \leq T_i < T_e$, to some observable metric space \mathcal{O} with the associated metric $\mathcal{D}_{\mathcal{O}}$. In this work we restrict our attention to traces which are continuous with respect to time. A *retiming* $r : [T_i, T_e] \rightarrow [T'_i, T'_e]$ is an order preserving continuous bijective function from $[T_i, T_e]$ to $[T'_i, T'_e]$; thus if $t < t'$ then $r(t) < r(t')$. Let the class of retiming functions from $[T_i, T_e]$ to $[T'_i, T'_e]$ be denoted as $R_{[T_i, T_e] \rightarrow [T'_i, T'_e]}$, and let I be the identity retiming. Intuitively, retiming can be thought of as follows: imagine a stretchable and compressible rubber bar; a retiming of the bar gives a configuration of the bar where some parts have been stretched, and some compressed, without the bar having been broken. Given a trace $x : [T_i^x, T_e^x] \rightarrow \mathcal{O}$, and a retiming $r : [T_i, T_e] \rightarrow [T'_i, T'_e]$; the function $x \circ r$ is another trace from $[T_i, T_e]$ to \mathcal{O} .

Definition 1 (Skorokhod Metric). Given a retiming $r : [T_i, T_e] \rightarrow [T'_i, T'_e]$, let $\|r - \text{I}\|_{\text{sup}}$ be defined as

$$\|r - \text{I}\|_{\text{sup}} = \sup_{t \in [T_i, T_e]} |r(t) - t|.$$

Given two traces $x : [T_i^x, T_e^x] \rightarrow \mathcal{O}$ and $y : [T_i^y, T_e^y] \rightarrow \mathcal{O}$, where \mathcal{O} is a metric space with the associated metric $\mathcal{D}_{\mathcal{O}}$, and a retiming $r : [T_i^x, T_e^x] \rightarrow [T_i^y, T_e^y]$, let $\|x - y \circ r\|_{\text{sup}}$ be defined as

$$\|x - y \circ r\|_{\text{sup}} = \sup_{t \in [T_i^x, T_e^x]} \mathcal{D}_{\mathcal{O}}(x(t), y(r(t))).$$

The *Skorokhod distance* between the traces $x()$ and $y()$ is defined to be:

$$S(x, y) = \inf_{r \in R_{[T_i^x, T_e^x] \rightarrow [T_i^y, T_e^y]}} \max(\|r - \text{I}\|_{\text{sup}}, \|x - y \circ r\|_{\text{sup}}).$$

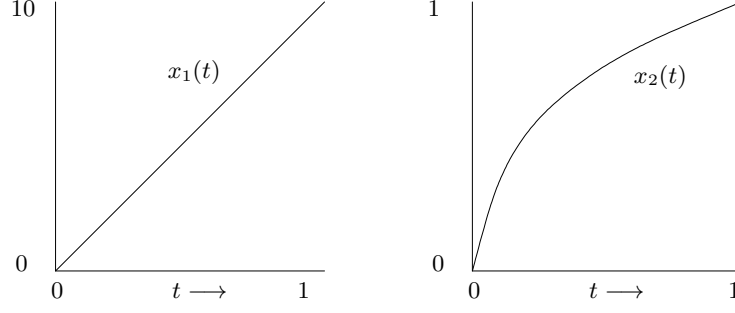


Figure 1: Original trace $x(t) = \langle x_1(t), x_2(t) \rangle$.

Intuitively, the Skorokhod distance incorporates two components: the first component quantifies the *timing discrepancy* of the timing distortion required to “match” two traces, and the second quantifies the *value mismatch* (in the metric space \mathcal{O}) of the values under the timing distortion. In the retimed trace $y \circ r$, we see exactly the same values as in y , in exactly the same order, but the times at which the value are seen can be different.

Remark 1. The two components of the Skorokhod distance (the retiming, and the value difference components) can be weighed with different weights – this simply corresponds to a change of scale.

Example 1 (Retimed Traces). We illustrate retimings and retimed traces in the next example. Let x be a trace $x : [0, 1] \mapsto \mathbb{R}^2$ defined by $x(t) = \langle 10 \cdot t, \sqrt{t} \rangle$. Consider a retiming $r : [0, 1] \mapsto [0, 1]$ defined by:

$$r(t) = \begin{cases} t^2 & \text{for } 0 \leq t \leq 1/2 \\ \frac{3}{2} \cdot t - \frac{1}{2} & \text{for } 1/2 \leq t \leq 1 \end{cases}$$

The maximum timing distortion of r , *i.e.*, $\sup_{t \in [0,1]} |r(t) - t|$, is $1/4$. The retimed trace $x(r(t))$ is given by $x(r(t)) = \langle x_1^r(t), x_2^r(t) \rangle$, where

$$x_1^r(t) = \begin{cases} 10 \cdot t^2 & \text{for } 0 \leq t \leq 1/2 \\ 10 \cdot (\frac{3}{2} \cdot t - \frac{1}{2}) & \text{for } 1/2 \leq t \leq 1 \end{cases}$$

and

$$x_2^r(t) = \begin{cases} t & \text{for } 0 \leq t \leq 1/2 \\ \sqrt{\frac{3}{2} \cdot t - \frac{1}{2}} & \text{for } 1/2 \leq t \leq 1. \end{cases}$$

The original and the retimed traces are depicted in Figures 1 and 2. □

Polygonal and Discrete-Time Traces. A polygonal trace $x : [T_i, T_e] \mapsto \mathcal{O}$ where \mathcal{O} is a vector space with the scalar field \mathbb{R} is a trace such that there exists a finite sequence $T_i = t_0 < t_1 < \dots < t_m = T_e$ such that the trace segment between t_k and t_{k+1} is affine for all $0 \leq k < m$, *i.e.* for $t_k \leq t \leq t_{k+1}$ we have $x(t) = x(t_k) + \frac{t-t_k}{t_{k+1}-t_k} \cdot (x(t_{k+1}) - x(t_k))$. Polygonal traces are obtained when discrete-time traces are completed by linear interpolation. A *discrete-time* trace \mathbf{x}_d is a finite sequence $\langle x_0, t_0 \rangle, \langle x_1, t_1 \rangle, \dots, \langle x_m, t_m \rangle$ such that for all $0 \leq k \leq m$, we have $x_k \in \mathcal{O}$; and that the t_k sequence is a strictly increasing sequence of time points with values in \mathbb{R}_+ . If \mathcal{O} is a vector space with the scalar field \mathbb{R} , the *linear interpolation trace* $\mathcal{Li}(\mathbf{x}_d)$ of the discrete time trace \mathbf{x}_d

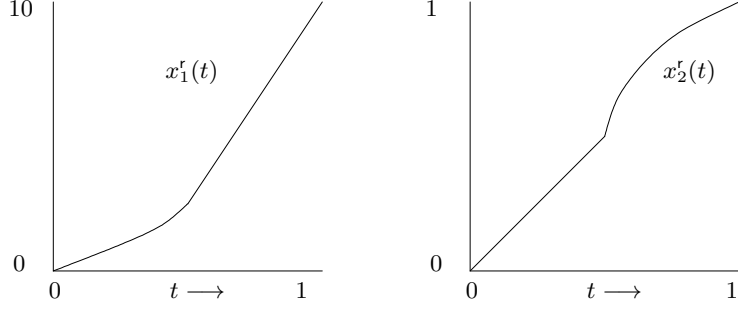


Figure 2: Retimed trace $x(r(t)) = \langle x_1^r(t), x_2^r(t) \rangle$.

is the continuous time polygonal trace $x()$ over $[t_0, t_n]$ defined as $x(t) = x_k + \lambda \cdot (x_{k+1} - x_k)$ for $t_k \leq t \leq t_{k+1}$ where $t = t_k + \lambda \cdot (t_{k+1} - t_k)$ and $0 \leq \lambda \leq 1$.

Continuous time Skorokhod Distance given Discrete-Time Traces. Given two discrete-time traces $\mathbf{x}_d, \mathbf{y}_d$ the (continuous time) Linear Interpolation Skorokhod distance between them is defined to be the Skorokhod distance between the polygonal traces obtained by linear interpolation.

$$\mathcal{S}_{\mathcal{Li}}(\mathbf{x}_d, \mathbf{y}_d) = \mathcal{S}(\mathcal{Li}(\mathbf{x}_d), \mathcal{Li}(\mathbf{y}_d))$$

That is, we assume that given the discrete-time samples of the traces, the actual continuous time traces are polygonal traces where the affine segments can be obtained by linear interpolation of the sampled values; the Skorokhod distance is computed for these polygonal traces.

Remark 2. Let $y() = \mathcal{Li}(\mathbf{y}_d)$ be a linear interpolation trace. We remark that after retiming, the retimed trace $y \circ r$ *need not* be piecewise linear. As an example, let $y : [0, 1] \rightarrow [0, 100]$ be a linear trace defined by $y(t) = 100 \cdot t$. Let the retiming $r : [0, 1] \rightarrow [0, 1]$ be defined by $r(t) = t^2$. Then $y \circ r$ is the trace $z : [0, 1] \rightarrow [0, 100]$ where $z(t) = 100 \cdot t^2$. Figure 3 illustrates four valid retimed traces from the original trace $y()$ in the center.

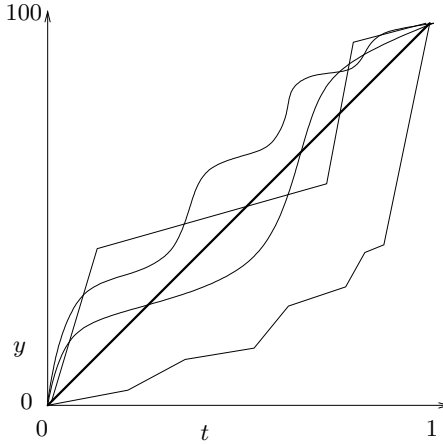


Figure 3: Retimed traces from $y(t) = 100 \cdot t$.

3 From Skorokhod to Fréchet Distances

We apply the work on Fréchet distances [AG95] towards computing Skorokhod distances between polygonal traces. The work of [AG95] deals with polygonal *curves*.

Definition 2 (Curves). A *curve* is a continuous mapping $f : [a, b] \rightarrow \mathcal{V}$ where $a, b \in \mathbb{R}_+$ with $a < b$, and \mathcal{V} is a vector space with the scalar field \mathbb{R} . A *polygonal curve* is a curve $P : [0, m] \rightarrow \mathcal{V}$ with $m \in \mathbb{N}$, such that for $0 \leq i < m$ the segment of P over $[i, i+1]$ is obtained by linear interpolation, i.e., $P(i + \lambda) = (1 - \lambda) \cdot P(i) + \lambda \cdot P(i + 1)$ for $0 \leq \lambda \leq 1$.

Given two points x, x' , we denote the affine segment between the two points as $\text{Line}(x, x')$.

Definition 3 (Fréchet distance). Let $f : [a_f, b_f] \rightarrow \mathcal{V}$ and $g : [a_g, b_g] \rightarrow \mathcal{V}$ be curves. The Fréchet distance between the two curves is defined to be

$$\mathcal{F}(f, g) = \inf_{\substack{\alpha_f : [0, 1] \rightarrow [a_f, b_f] \\ \alpha_g : [0, 1] \rightarrow [a_g, b_g]}} \max_{0 \leq \theta \leq 1} \|f(\alpha_f(\theta)) - g(\alpha_g(\theta))\|$$

where $\|\cdot\|$ is the norm over \mathcal{V} , and where α_f, α_g range over continuous and strictly increasing bijective functions onto $[a_f, b_f]$ and $[a_g, b_g]$ respectively.

Intuitively, the reparameterizations α_f, α_g control the “speed” of traversal along the two curves by two entities. The positions of the two entities in the two curves at “time” θ is given by $\alpha_f(\theta)$ and $\alpha_g(\theta)$ respectively; with the value of the curves at those positions being $f(\alpha_f(\theta))$ and $g(\alpha_g(\theta))$ respectively. The two entities always have a strictly greater than 0 speed.

Observe that unlike the Skorokhod distance, the Fréchet distance does not impose a penalty for the amount of position mismatch, in particular, almost the whole curve g can be matched to a tiny portion of f , with no penalty. We show that, still, the Skorokhod distance between continuous traces can be obtained using the Fréchet distance between two derived curves. First we obtain a curve given a continuous time trace.

Definition 4 (Curve of a Trace). Let \mathcal{O} be a vector space with the scalar field \mathbb{R} . Let $x : [T_i^x, T_e^x] \rightarrow \mathcal{O}$ be a continuous trace. The *curve* of the trace $x()$ is the curve $C_x : [T_i^x, T_e^x] \mapsto \mathcal{O} \times [T_i^x, T_e^x]$ defined by the following parameterization. For $\rho \in [T_i^x, T_e^x]$ we define (a) $C_x^{\mathcal{O}}(\rho) = x(\rho)$; and (b) $C_x^{\mathbb{R}}(\rho) = \rho$. The curve C_x is given by $\langle C_x^{\mathcal{O}}(\rho), C_x^{\mathbb{R}}(\rho) \rangle$ over $[T_i^x, T_e^x]$ with values in $\mathcal{O} \times \mathbb{R}$. We defined the norm of the space $\mathcal{O} \times \mathbb{R}$ as $\|\langle o, t \rangle\| = \max(\|o\|, |t|)$. We refer to this norm as the $L_{\mathcal{O}}^{\mathbb{S}}$ norm, where $L_{\mathcal{O}}$ is the norm of \mathcal{O} , and to the space $\mathcal{O} \times \mathbb{R}$ as the $\mathcal{O}^{\mathbb{S}}$ space.

We note that $\langle C_x^{\mathcal{O}}(\rho), C_x^{\mathbb{R}}(\rho) \rangle$ is a continuous mapping from $[T_i^x, T_e^x]$ to $\mathcal{O} \times [T_i^x, T_e^x]$ since $x()$ is a continuous trace. The next lemma shows that $L_{\mathcal{O}}^{\mathbb{S}}$ is a norm. Thus, the previous definition is sound.

Lemma 1. Let \mathcal{O} be a vector space with the scalar field \mathbb{R} . Consider the space $\mathcal{O} \times \mathbb{R}$. The function $L_{\mathcal{O}}^{\mathbb{S}} : \mathcal{O} \times \mathbb{R} \mapsto \mathbb{R}_+$ defined by $L_{\mathcal{O}}^{\mathbb{S}}(\langle p, t \rangle) = \max(\|p\|_{\mathcal{O}}, |t|)$ is a norm on $\mathcal{O} \times \mathbb{R}$.

Proof. It can be easily seen that $L_{\mathcal{O}}^{\mathbb{S}}(\langle a \cdot p, a \cdot t \rangle) = |a| \cdot L_{\mathcal{O}}^{\mathbb{S}}(\langle p, t \rangle)$ for $a \in \mathbb{R}$, and $L_{\mathcal{O}}^{\mathbb{S}}(\langle p, t \rangle) = 0$ iff $p = \vec{0}$

and $t = 0$, where $\vec{0}$ is the zero vector of \mathcal{O} . We now check for the triangle inequality.

$$\begin{aligned} L_{\mathcal{O}}^{\mathcal{S}}(\langle p, t_p \rangle) + L_{\mathcal{O}}^{\mathcal{S}}(\langle q, t_q \rangle) &= \max(\|p\|_{\mathcal{O}}, |t_p|) + \max(\|q\|_{\mathcal{O}}, |t_q|) \\ &\geq \frac{\|p\|_{\mathcal{O}} + \|q\|_{\mathcal{O}}}{|t_p| + |t_q|}; \text{ and } \text{both.} \end{aligned}$$

$$\begin{aligned} \text{Thus, } L_{\mathcal{O}}^{\mathcal{S}}(\langle p, t_p \rangle) + L_{\mathcal{O}}^{\mathcal{S}}(\langle q, t_q \rangle) &\geq \max(\|p\|_{\mathcal{O}} + \|q\|_{\mathcal{O}}, |t_p| + |t_q|) \\ &\geq \max(\|p + q\|_{\mathcal{O}}, |t_p + t_q|) \\ &= L_{\mathcal{O}}^{\mathcal{S}}(\langle p + q, t_p + t_q \rangle) \\ &= L_{\mathcal{O}}^{\mathcal{S}}(\langle p, t_p \rangle + \langle q, t_q \rangle) \end{aligned}$$

Thus the triangle inequality holds and this concludes the proof. \square

The next result shows that the Skorokhod distance between two continuous traces is equal to the Fréchet distance between two related curves in a corresponding normed space.

Proposition 1 (From Skorokhod to Fréchet). *Let \mathcal{O} be a vector space with the scalar field \mathbb{R} . Let $x : [T_i^x, T_e^x] \rightarrow \mathcal{O}$ and $y : [T_i^y, T_e^y] \rightarrow \mathcal{O}$ be two continuous traces. We have*

$$\mathcal{S}(x, y) = \mathcal{F}(\mathbf{C}_x, \mathbf{C}_y)$$

where the curves \mathbf{C}_x and \mathbf{C}_y are the curves in the space $\mathcal{O}^{\mathcal{S}}$ corresponding to the traces x and y as defined in Definition 4.

Proof. We prove $\mathcal{S}(x, y) \leq \mathcal{F}(\mathbf{C}_x, \mathbf{C}_y)$, and that $\mathcal{F}(\mathbf{C}_x, \mathbf{C}_y) \leq \mathcal{S}(x, y)$. Let $f = \mathbf{C}_x$ and $g = \mathbf{C}_y$.

1. $\mathcal{S}(x, y) \leq \mathcal{F}(\mathbf{C}_x, \mathbf{C}_y)$. Let α_f, α_g be the reparametrizations as in Definition 3. Consider a retiming $\mathbf{r} : [T_i^x, T_e^x] \rightarrow [T_i^y, T_e^y]$ defined as $\mathbf{r}(t) = \alpha_g(\alpha_f^{-1}(t))$. It can be checked that the function as defined is a valid retiming. Now,

$$\begin{aligned} \max(\|\mathbf{r} - \mathbf{I}\|_{\sup}, \|x - y \circ \mathbf{r}\|_{\sup}) &= \max(\|\mathbf{r} - \mathbf{I}\|_{\sup}, \max_{T_i^x \leq t \leq T_e^x} \|x(t) - y(\alpha_g(\alpha_f^{-1}(t)))\|) \\ &= \max\left(\|\mathbf{r} - \mathbf{I}\|_{\sup}, \max_{0 \leq \theta \leq 1} \|x(\alpha_f(\theta)) - y(\alpha_g(\theta))\|\right) \end{aligned}$$

by setting $t = \alpha_f(\theta)$ in the last equation.

$$\begin{aligned} &= \max\left(\max_{T_i^x \leq t \leq T_e^x} |\alpha_g(\alpha_f^{-1}(t)) - t|, \max_{0 \leq \theta \leq 1} \|x(\alpha_f(\theta)) - y(\alpha_g(\theta))\|\right) \\ &= \max\left(\max_{0 \leq \theta \leq 1} |\alpha_g(\theta) - \alpha_f(\theta)|, \max_{0 \leq \theta \leq 1} \|x(\alpha_f(\theta)) - y(\alpha_g(\theta))\|\right) \\ &= \max_{0 \leq \theta \leq 1} \max(|\alpha_g(\theta) - \alpha_f(\theta)|, \|x(\alpha_f(\theta)) - y(\alpha_g(\theta))\|) \end{aligned}$$

by interchanging the order of taking maximums

in the previous equation.

$$\begin{aligned} &= \max_{0 \leq \theta \leq 1} \|\langle x(\alpha_f(\theta)) - y(\alpha_g(\theta)), \alpha_f(\theta) - \alpha_g(\theta) \rangle\| \\ &= \max_{0 \leq \theta \leq 1} \|f(\alpha_f(\theta)) - g(\alpha_g(\theta))\| \end{aligned}$$

Thus, for every valid choice of α_f, α_g , there is a valid retiming function r such that $\max(\|r - I\|_{\text{sup}}, \|x - y \circ r\|_{\text{sup}}) = \max_{0 \leq \theta \leq 1} \|f(\alpha_f(\theta)) - g(\alpha_g(\theta))\|$. Hence $\mathcal{S}(x, y) \leq \mathcal{F}(\mathbb{C}_x, \mathbb{C}_y)$.

2. $\mathcal{S}(x, y) \geq \mathcal{F}(\mathbb{C}_x, \mathbb{C}_y)$.

Let $r : [T_i^x, T_e^x] \rightarrow [T_i^y, T_e^y]$ be a retiming. Let $\alpha_f : [0, 1] \rightarrow [T_i^x, T_e^x]$ be defined as $\alpha_f(\theta) = (1 - \theta) \cdot T_i^x + \theta \cdot T_e^x$. Let $\alpha_g : [0, 1] \rightarrow [T_i^y, T_e^y]$ be defined as $\alpha_g(\theta) = r((1 - \theta) \cdot T_i^x + \theta \cdot T_e^x)$. Thus, $\alpha_g(\theta) = r(\alpha_f(\theta))$. Observe that α_f, α_g satisfy the conditions of Definition 3. We have,

$$\begin{aligned} \max_{0 \leq \theta \leq 1} \|f(\alpha_f(\theta)) - g(\alpha_g(\theta))\| &= \max_{0 \leq \theta \leq 1} \max(|\alpha_f(\theta) - \alpha_g(\theta)|, \|x(\alpha_f(\theta)) - y(\alpha_g(\theta))\|) \\ &= \max \left(\max_{0 \leq \theta \leq 1} |\alpha_f(\theta) - \alpha_g(\theta)|, \max_{0 \leq \theta \leq 1} \|x(\alpha_f(\theta)) - y(\alpha_g(\theta))\| \right) \\ &\text{by interchanging the order of taking maximums} \\ &\text{in the previous equation.} \\ &= \max \left(\max_{T_i^x \leq t \leq T_e^x} |t - r(t)|, \max_{T_i^x \leq t \leq T_e^x} \|x(t) - y(r(t))\| \right) \\ &\text{by setting } t = \alpha_f(\theta) \text{ in the last equation.} \\ &= \max(\|I - r\|_{\text{sup}}, \|x - y \circ r\|_{\text{sup}}) \end{aligned}$$

Thus, for every valid retiming function r , there is a valid choice of α_f, α_g such that $\max(\|r - I\|_{\text{sup}}, \|x - y \circ r\|_{\text{sup}}) = \max_{0 \leq \theta \leq 1} \|f(\alpha_f(\theta)) - g(\alpha_g(\theta))\|$. Hence $\mathcal{F}(\mathbb{C}_x, \mathbb{C}_y) \leq \mathcal{S}(x, y)$. □

4 Computation of the Fréchet Distance

In this section we explore in detail the algorithm sketch of [AG95] for computing Fréchet distances in \mathbb{R}^2 for the L_2 norm (see also [Kna02] and [Wen03]). We derive, and prove where necessary, missing results, and generalizations of the results which are needed in our algorithm for computing the Skorokhod distances in higher dimensions for the L_1 , L_2 , and L_∞ metrics. We first solve for the *decision problem* in \mathbb{R}^n for the different norms L_1, L_2 and L_∞ , and also for L_1^S, L_2^S and L_∞^S (the three new norms are required for solving the Skorokhod distance problem in the three standard norms). We then solve for the *value computation* problem by obtaining a geometric characterization for quantities which are necessary for computing the value of the Fréchet distance. Using this geometric characterization, the desired quantities required in the Fréchet distance are computed by solving geometric problems for the various norms in Section 5.

4.1 The Free Space

This subsection present the notion of *Free Space* from [AG95] and its characterization for L_1, L_2, L_∞ and also for L_1^S, L_2^S and L_∞^S norms in the \mathbb{R}^n space. The Free Space concept allows us to reduce reasoning about the Fréchet distance in any dimension to reasoning about paths in \mathbb{R}^2 .

Given a polygonal curve $f : [0, n] \rightarrow \mathcal{V}$, we let $f_{[i]}$ denote the curve segment between $[i, i + 1]$. Thus, $f_{[i]} : [i, i + 1] \rightarrow \mathcal{V}$ and $f(\rho) = f_{[i]}(\rho)$ for $i \leq \rho \leq i + 1$.

Definition 5 (Free Space [AG95]). Given polygonal curves $f : [0, m_f] \rightarrow \mathcal{V}$ and $g : [0, m_g] \rightarrow \mathcal{V}$, and a real number $\delta \geq 0$, δ -Free Space of f, g is the set

$$\text{Free}_\delta(f, g) = \{(\rho_f, \rho_g) \in [0, m_f] \times [0, m_g] \mid \text{we have } \|f(\rho_f) - g(\rho_g)\|_{\mathcal{V}} \leq \delta\}$$

The tuples (ρ_f, ρ_g) belonging to $\text{Free}_\delta(f, g)$ denote the positions in the two curves such that the difference in the values of the two curves is less than δ . A pictorial representation of the free space is referred to as the *free space diagram*. The space $[0, m_f] \times [0, m_g]$ can be viewed as consisting of $m_f \cdot m_g$ cells, with cell i, j being $[i, i+1] \times [j, j+1]$ for $0 \leq i < m_f$, and $0 \leq j < m_g$. Observe that $\text{Free}_\delta(f, g)$ intersected with cell i, j is just the free space corresponding to the curve segments $f_{[i]}, g_{[j]}$; i.e the intersection of the cell i, j with $\text{Free}_\delta(f, g)$ is equal to $\text{Free}_\delta(f_{[i]}, g_{[j]})$.

Proposition 2 ([AG95]). Given two polygonal curves f, g , we have $\mathcal{F}(f, g) \leq \delta$ if there is a curve $\alpha : [0, 1] \rightarrow [0, m_f] \times [0, m_g]$ in $\text{Free}_\delta(f, g)$ from $(0, 0)$ to (m_f, m_g) which is strictly increasing in both coordinates¹.

The curve α can be thought of as the parameterized curve (α_f, α_g) , with $\alpha_f : [0, 1] \rightarrow [0, m_f]$, $\alpha_g : [0, 1] \rightarrow [0, m_g]$. These functions α_f, α_g can be viewed as the reparametrization functions in Definition 3. An example of the free space for two polygonal curves is given in Figure 4. The unshaded portion is the free space. The figure also includes a continuous curve which is strictly increasing in both coordinates, from $(0, 0)$ to (m_f, m_g) . We now analyze the properties of $\text{Free}_\delta(f_{[i]}, g_{[j]})$.

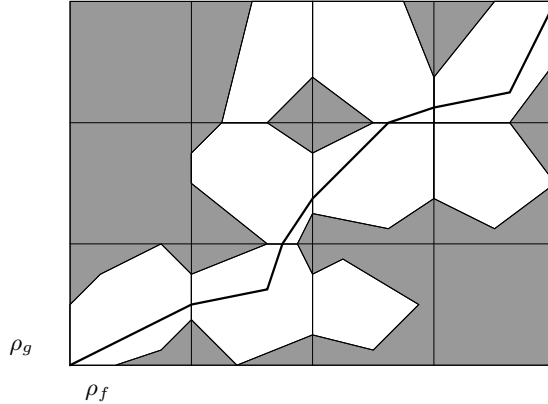


Figure 4: The Free Space $\text{Free}_\delta(f, g)$.

Definition 6. For $(\rho_f, \rho_g) \in [i, i+1] \times [j, j+1]$, we denote $\rho_f^* = \rho_f - i$, and $\rho_g^* = \rho_g - j$, i.e., we move the origin to (i, j) . \square

Proposition 3 (Free Space for Affine Line Segments in \mathbb{R}^1). Let $T_x : [i, i+1] \rightarrow \mathbb{R}$ and $T_y : [j, j+1] \rightarrow \mathbb{R}$ be affine line segments. The set $\text{Free}_\delta(T_x, T_y)$ for the absolute value norm is a

¹The corresponding result (stated without proof) in [AG95] also includes the “only if” direction. However, [AG95] only requires *non-decreasing* reparametrizations, as opposed to our formulation which requires strictly increasing reparametrizations necessary for utilizing Fréchet distances in computing Skorokhod distances. A careful analysis shows that the “only if” direction of the proposition also holds in case of non-decreasing reparametrizations, but not for strictly increasing reparametrizations. We address this issue in detail later in the paper.

portion of $[i, i+1] \times [j, j+1]$ which lies in the intersection of two half-planes, the boundaries of which are parallel.

$$\mathsf{T}_x(i) - \mathsf{T}_y(j) + (\mathsf{T}_x(i+1) - \mathsf{T}_x(i)) \cdot \rho_x^* + (\mathsf{T}_y(j) - \mathsf{T}_y(j+1)) \cdot \rho_y^* \leq \delta \quad (1)$$

$$\mathsf{T}_x(i) - \mathsf{T}_y(j) + (\mathsf{T}_x(i+1) - \mathsf{T}_x(i)) \cdot \rho_x^* + (\mathsf{T}_y(j) - \mathsf{T}_y(j+1)) \cdot \rho_y^* \geq -\delta \quad (2)$$

Proof. We have $\mathsf{T}_x(i + \rho_x) = (1 - \rho_x) \cdot \mathsf{T}_x(i) + \rho_x \cdot \mathsf{T}_x(i+1)$ for $0 \leq \rho_x \leq 1$; and similarly $\mathsf{T}_y(j + \rho_y) = (1 - \rho_y) \cdot \mathsf{T}_y(j) + \rho_y \cdot \mathsf{T}_y(j+1)$ for $0 \leq \rho_y \leq 1$. Thus,

$$\mathsf{T}_x(i + \rho_x) - \mathsf{T}_y(j + \rho_y) = \mathsf{T}_x(i) - \mathsf{T}_y(j) + \rho_x \cdot (\mathsf{T}_x(i+1) - \mathsf{T}_x(i)) - \rho_y \cdot (\mathsf{T}_y(j+1) - \mathsf{T}_y(j))$$

We have $|\mathsf{T}_x(i + \rho_x) - \mathsf{T}_y(j + \rho_y)| \leq \delta$ iff $\mathsf{T}_x(i + \rho_x) - \mathsf{T}_y(j + \rho_y) \leq \delta$ and $\mathsf{T}_y(j + \rho_y) - \mathsf{T}_x(i + \rho_x) \leq \delta$. Thus, the points $(i + \rho_x, j + \rho_y)$ which are in the intersection of the two half-planes of the proposition constitute the set $\text{Free}_\delta(\mathsf{T}_x, \mathsf{T}_y)$. We conclude the proof by noting that the two lines denoting the half-plane boundaries are parallel. \square

Proposition 4 (Convexity of Free Space). *Let $f_{[i]} : [i, i+1] \rightarrow \mathbb{R}^n$ and $g_{[j]} : [j, j+1] \rightarrow \mathbb{R}^n$ be straight line segments of two polygonal curves. Given any $\delta \geq 0$, the set $\text{Free}_\delta(f_{[i]}, g_{[j]})$ is convex for any norm.*

Proof. Let $0 \leq \lambda \leq 1$, and let $(\rho_f^0, \rho_g^0) \in [i, i+1] \times [j, j+1]$ and $(\rho_f^1, \rho_g^1) \in [i, i+1] \times [j, j+1]$ be in $\text{Free}_\delta(f, g)$. Consider $(1 - \lambda) \cdot (\rho_f^0, \rho_g^0) + \lambda \cdot (\rho_f^1, \rho_g^1)$.

We have $\|f_{[i]}((1 - \lambda) \cdot \rho_f^0 + \lambda \cdot \rho_f^1) - g_{[j]}((1 - \lambda) \cdot \rho_g^0 + \lambda \cdot \rho_g^1)\| =$

$$\left\| (1 - \lambda) \cdot f_{[i]}(\rho_f^0) + \lambda \cdot f_{[i]}(\rho_f^1) - \left((1 - \lambda) \cdot g_{[j]}(\rho_g^0) + \lambda \cdot g_{[j]}(\rho_g^1) \right) \right\|$$

Since $f_{[i]}$ and $g_{[j]}$ are line segments containing

$f_{[i]}(\rho_f^0)$ and $f_{[i]}(\rho_f^1)$; and $g_{[j]}(\rho_g^0)$ and $g_{[j]}(\rho_g^1)$ respectively.

$$\begin{aligned} &= \left\| (1 - \lambda) \cdot (f_{[i]}(\rho_f^0) - g_{[j]}(\rho_g^0)) + \lambda \cdot (f_{[i]}(\rho_f^1) - g_{[j]}(\rho_g^1)) \right\| \\ &\leq (1 - \lambda) \cdot \|f_{[i]}(\rho_f^0) - g_{[j]}(\rho_g^0)\| + \lambda \cdot \|f_{[i]}(\rho_f^1) - g_{[j]}(\rho_g^1)\| \\ &\leq (1 - \lambda) \cdot \delta + \lambda \cdot \delta \end{aligned}$$

Thus, $(1 - \lambda) \cdot (\rho_f^0, \rho_g^0) + \lambda \cdot (\rho_f^1, \rho_g^1)$ belongs to $\text{Free}_\delta(f_{[i]}, g_{[j]})$. \square

4.2 Characterizing the Free Space for Different Norms

We give characterizations of $\text{Free}_\delta(f_{[i]}, g_{[j]})$ for affine line segments $f_{[i]}, g_{[j]}$ for the three standard norms L_1, L_2 , and L_∞ , as well as for the three other norms L_1^S, L_∞^S and L_2^S .

Recall that the L_1 norm defines $\|(d_1, \dots, d_n)\|_{L_1}$ to be $\sum_{k=1}^n |d_k|$; the L_2 norm defines $\|(d_1, \dots, d_n)\|_{L_2}$ to be $\sqrt{\sum_{k=1}^n d_k^2}$; and the L_∞ norm defines $\|(d_1, \dots, d_n)\|_{L_\infty}$ to be $\max_k |d_k|$. The L_1^S, L_∞^S and L_2^S norms are obtained from these three standard norms as defined in Definition 4, thus, $\|\langle o, t \rangle\|_{\chi^S} = \max(\|o\|_\chi, |t|)$ for $\chi \in \{L_1, L_2, L_\infty\}$. If f is a curve $f : [a_f, b_f] \rightarrow \mathbb{R}^n$, we denote by f_k the k -th dimension component of f for $0 \leq k \leq n$. Note that f_k is also a curve

$f_k : [a_f, b_f] \rightarrow \mathbb{R}$ In the following we use the following two affine identities for the affine portions of polygonal curves. For $(\rho_f, \rho_g) \in [i, i+1] \times [j, j+1]$, and $(\rho_f^*, \rho_g^*) = (\rho_f - i, \rho_g - j)$ we have:

$$\begin{aligned} f(\rho_f^*) &= f(i) + (f(i+1) - f(i)) \cdot \rho_f^* \\ g(\rho_g^*) &= g(j) + (g(j+1) - g(j)) \cdot \rho_g^* \end{aligned} \quad (3)$$

Proposition 5 (Free Space for L_∞). *Let $f : [a_f, b_f] \rightarrow \mathbb{R}^n$ and $g : [a_g, b_g] \rightarrow \mathbb{R}^n$ be curves. If the space \mathbb{R}^n has the L_∞ norm, then given a $\delta \geq 0$ we have $\text{Free}_\delta(f, g) = \cap_{k=1}^m \text{Free}_\delta(f_k, g_k)$, where f_k, g_k are the k -th dimension component curves of f, g .*

Proof. We have $(\rho_f, \rho_g) \in \text{Free}_\delta(f, g)$ iff $\max_k |f_k(\rho_f) - g_k(\rho_g)| \leq \delta$. This holds iff $|f_k(\rho_f) - g_k(\rho_g)| \leq \delta$ for all $0 \leq k \leq n$. The result follows. \square

Proposition 6 (Polygonal Free Space for L_∞). *Let $f : [0, m_f] \rightarrow \mathbb{R}^n$ and $g : [0, m_g] \rightarrow \mathbb{R}^n$ be polygonal curves. If the space \mathbb{R}^n has the L_∞ norm, then given a $\delta \geq 0$ we have $\text{Free}_\delta(f, g) \cap ([i, i+1] \times [j, j+1])$ for $i < m_f$ and $j < m_g$ to be the intersection of the regions between n pairs of parallel lines. Each such region is the intersection of the following two half planes for $1 \leq k \leq m$, and $(\rho_f^*, \rho_g^*) \in [0, 1]^2$:*

$$f_k(i) - g_k(j) + (f_k(i+1) - f_k(i)) \cdot \rho_x^* + (g_k(j) - g_k(j+1)) \cdot \rho_y^* \leq \delta \quad (4)$$

$$f_k(i) - g_k(j) + (f_k(i+1) - f_k(i)) \cdot \rho_x^* + (g_k(j) - g_k(j+1)) \cdot \rho_y^* \geq -\delta \quad (5)$$

Proof. The result follows using similar ideas as in the proofs of Proposition 5 and Proposition 3. \square

Proposition 7 (Polygonal Free Space for L_1 norm). *Let $f : [0, m_f] \rightarrow \mathbb{R}^n$ and $g : [0, m_g] \rightarrow \mathbb{R}^n$ be polygonal curves. If the space \mathbb{R}^n has the L_∞ norm, then given a $\delta \geq 0$ we have $\text{Free}_\delta(f, g) \cap ([i, i+1] \times [j, j+1])$ for $i < m_f$ and $j < m_g$ to be the part of a polytope, formed by the intersection of the following 2^n half-planes $H_{(q_1, \dots, q_n)}$, which lies inside $[i, i+1] \times [j, j+1]$:*

$$\sum_{k=1}^n (-1)^{q_k} \cdot \left(f_k(i) - g_k(j) + (f_k(i+1) - f_k(i)) \cdot \rho_f^* + (g_k(j) - g_k(j+1)) \cdot \rho_g^* \right) \leq \delta$$

for $(q_1, \dots, q_n) \in \{1, 2\}^n$; or equivalently the region of points $(\rho_f, \rho_g) \in [i, i+1] \times [j, j+1]$ satisfying the following inequality:

$$\sum_{k=1}^n \left| \left(f_k(i) - g_k(j) + (f_k(i+1) - f_k(i)) \cdot \rho_f^* + (g_k(j) - g_k(j+1)) \cdot \rho_g^* \right) \right| \leq \delta$$

where $(\rho_f^*, \rho_g^*) = (\rho_f - i, \rho_g - j)$.

Proof. We have

$$(\rho_f, \rho_g) \in \text{Free}_\delta(f, g) \text{ iff } \sum_{k=1}^n |f_k(\rho_f) - g_k(\rho_g)| \leq \delta.$$

Let $f_k(\rho_f) - g_k(\rho_g) = d_k$. Let $q^*(d_k) = 1$ if $d_k \geq 0$ and -1 otherwise. Then we have

$$\sum_{k=1}^n |d_k| \leq \delta \text{ iff } \sum_{k=1}^n q^*(d_k) \cdot d_k \leq \delta. \quad (6)$$

Now observe the following. For any $(q_1, \dots, q_n) \in \{1, 2\}^n$, we have

$$\sum_{k=1}^n (-1)^{q_k} \cdot d_k \leq \sum_{k=1}^n q^*(d_k) \cdot d_k. \quad (7)$$

This is because for any $q_k \in \{1, 2\}$, we have $(-1)^{q_k} \cdot d_k \leq q^*(d_k) \cdot d_k$. Moreover, we have that there exists a $(q_1, \dots, q_n) \in \{1, 2\}^n$ such that an exact inequality holds in Equation 7. Using these facts, we have that Equation 6 holds iff the following 2^n equations hold, one for each choice of $(q_1, \dots, q_n) \in \{1, 2\}^n$:

$$\sum_{k=1}^n (-1)^{q_k} \cdot (f_k(\rho_f) - g_k(\rho_g)) \leq \delta$$

The result follows using Equations 3 □

Proposition 8 (Polygonal Free Space for L_2). *Let $f : [0, m_f] \rightarrow \mathbb{R}^n$ and $g : [0, m_g] \rightarrow \mathbb{R}^n$ be polygonal curves. If the space \mathbb{R}^n has the L_2 norm, then given a $\delta \geq 0$ we have the boundary of the free space $\text{Free}_\delta(f, g) \cap ([i, i+1] \times [j, j+1])$ for $i < m_f$ and $j < m_g$ to be the intersection of an ellipse, or a parabola, with $[i, i+1] \times [j, j+1]$.*

Proof. We have $(\rho_f, \rho_g) \in \text{Free}_\delta(f, g)$ iff

$$\sum_{k=1}^m (f(\rho_f) - g(\rho_g))^2 \leq \delta^2$$

Using Equations 3, the above is equivalent to

$$\sum_{k=1}^n \left(f_k(i) - g_k(j) + (f_k(i+1) - f_k(i)) \cdot \rho_f^* + (g_k(j) - g_k(j+1)) \cdot \rho_g^* \right)^2 \leq \delta^2$$

Let $d_k^0 = f_k(i) - g_k(j)$; $d_k^f = f_k(i+1) - f_k(i)$; and $d_k^g = g_k(j) - g_k(j+1)$. The previous equation can then be written as

$$\sum_{k=1}^n \left(d_k^0 + d_k^f \cdot \rho_f^* + d_k^g \cdot \rho_g^* \right)^2 \leq \delta^2$$

Expanding the above, we get

$$\sum_{k=1}^n \left((d_k^0)^2 + (d_k^f)^2 \cdot (\rho_f^*)^2 + (d_k^g)^2 \cdot (\rho_g^*)^2 + 2 \cdot \left(d_k^0 \cdot d_k^f \cdot \rho_f^* + d_k^f \cdot d_k^g \cdot \rho_f^* \cdot \rho_g^* + d_k^0 \cdot d_k^g \cdot \rho_g^* \right) \right) \leq \delta^2$$

Expanding the summation and rearranging, we get

$$\begin{aligned} \left(\sum_{k=1}^n (d_k^f)^2 \right) \cdot (\rho_f^*)^2 + \left(\sum_{k=1}^n (d_k^g)^2 \right) \cdot (\rho_g^*)^2 + \left(\sum_{k=1}^n 2 \cdot d_k^f \cdot d_k^g \right) \cdot \rho_f^* \cdot \rho_g^* + \left(\sum_{k=1}^n 2 \cdot d_k^0 \cdot d_k^f \right) \cdot \rho_f^* + \\ \left(\sum_{k=1}^n 2 \cdot d_k^0 \cdot d_k^g \right) \cdot \rho_g^* + \sum_{k=1}^n (d_k^0)^2 - \delta^2 \leq 0 \end{aligned} \quad (8)$$

Recall that the type of the conic $Ax^2 + Bxy + C^2y + Fx + Gy + H = 0$ depends on the sign of $B^2 - 4AC$ (see *e.g.* [DEG12]). This difference in our case is

$$\left(\sum_{k=1}^n 2 \cdot d_k^f \cdot d_k^g \right)^2 - 4 \cdot \left(\sum_{k=1}^n (d_k^f)^2 \right) \cdot \left(\sum_{k=1}^n (d_k^g)^2 \right)$$

To simplify the notation, we let $d_k^f = x_k$, and $d_k^g = y_k$. The above term is then, dropping the factor of 4,

$$\left(\sum_{k=1}^n x_k \cdot y_k \right)^2 - \left(\sum_{k=1}^n x_k^2 \right) \cdot \left(\sum_{k=1}^n y_k^2 \right)$$

This term is ≤ 0 by the Cauchy-Schwartz inequality. Thus, the boundary of the free space inside a cell is either an ellipse, or a parabola (it is a parabola iff there is constant P such that for all k we have $d_k^g = P \cdot d_k^f$). The equation of the boundary inside the cell is Equation 8. \square

Proposition 9 (Free Space for L_1^S, L_2^S, L_∞^S norms). *Let $f : [a_f, b_f] \rightarrow \mathbb{R}^n$ and $f_{n+1} : [a_f, b_f] \rightarrow \mathbb{R}$, and $g : [a_g, b_g] \rightarrow \mathbb{R}^n$ and $g : [a_g, b_g] \rightarrow \mathbb{R}$ be curves. Note that $\langle f, f_{n+1} \rangle$ is a curve from $[a_f, b_f]$ to $\mathbb{R}^n \times \mathbb{R}$. If the space $\mathbb{R}^n \times \mathbb{R}$ has the norm χ^S for $\chi \in \{L_1, L_2, L_\infty\}$, then given a $\delta \geq 0$ we have*

$$\text{Free}_\delta^{\chi^S}(\langle f, f_{n+1} \rangle, \langle g, g_{n+1} \rangle) = \text{Free}_\delta^\chi(f, g) \cap \text{Free}_\delta^{L_1}(f_{n+1}, g_{n+1})$$

where $\text{Free}_\delta^\chi()$ denote the free space for norm χ .

Proof. We have $(\rho_f, \rho_g) \in \text{Free}_\delta(f, g)$ iff $\max\left(\|f(\rho_f) - g(\rho_g)\|_\chi, |f_{n+1}(\rho_f) - g_{n+1}(\rho_g)|\right) \leq \delta$. This holds iff both $\|f(\rho_f) - g(\rho_g)\|_\chi \leq \delta$ and $|f_{n+1}(\rho_f) - g_{n+1}(\rho_g)| \leq \delta$. The result follows. \square

4.3 Computing Free Space Cell Boundaries

The algorithm for computing the Fréchet distance requires computation of the free space at the cell boundaries, *i.e.* $(\{i\} \times [j, j+1]) \cap \text{Free}_\delta(f, g)$ for $0 \leq i \leq m_f$ and $0 \leq j \leq m_g - 1$; and $([i, i+1] \times \{j\}) \cap \text{Free}_\delta(f, g)$ for $0 \leq i \leq m_f - 1$ and $0 \leq j \leq m_g$. Using Proposition 4, we get that thus the free space at the cell boundaries is also convex, and hence just a line. Hence, it suffices to just compute maximum and minimum coordinate values for the free space at the cell boundaries. We do this for the different norms.

Computing Free Space Cell Boundaries for L_1 . Recall from Proposition 7 that the free space for for cell i, j for $0 \leq i \leq m_f - 1$ and $0 \leq j \leq m_g - 1$ is given by the inequality

$$\sum_{k=1}^n \left| \left(f_k(i) - g_k(j) + (f_k(i+1) - f_k(i)) \cdot \rho_f^* + (g_k(j) - g_k(j+1)) \cdot \rho_g^* \right) \right| \leq \delta$$

where $(\rho_f^*, \rho_g^*) = (\rho_f - i, \rho_g - j)$, with $i \leq \rho_f \leq i+1$ and $j \leq \rho_g \leq j+1$. Letting $r_k^0 = f_k(i) - g_k(j)$, and $r_k^f = f_k(i+1) - f_k(i)$, and $r_k^g = g_k(j) - g_k(j+1)$, and $x = \rho_f^*$, and $y = \rho_g^*$, we obtain the inequality

$$\sum_{k=1}^n \left| r_k^0 + r_k^f \cdot x + r_k^g \cdot y \right| \leq \delta \tag{9}$$

with $0 \leq x \leq 1$, and $0 \leq y \leq 1$.

We show how to obtain the free space boundary at the cell boundary $0 \leq x \leq 1, y = 1$. The other cases are similar. We need to compute the maximum, and the minimum values of $0 \leq x \leq 1$ such that Equation 9 holds with $y = 1$. Substituting $y = 1$, and letting $r_k^1 = r_k^0 - r_k^g$, we thus, wish to solve the following two optimization problems.

$$\begin{aligned} & \text{minimize } x \\ & \text{subject to } \sum_{k=1}^n \left| r_k^1 + r_k^f \cdot x \right| \leq \delta \\ & 0 \leq x \leq 1 \end{aligned}$$

and

$$\begin{aligned} & \text{maximize } x \\ & \text{subject to } \sum_{k=1}^n \left| r_k^1 + r_k^f \cdot x \right| \leq \delta \\ & 0 \leq x \leq 1 \end{aligned}$$

where r_k^1, r_k^f , and δ are given constants. We show how to solve the maximization problem (the minimization problem is similar). Assume none of r_k^f is zero (if some r_k^f is zero, remove $|r_k^1 + r_k^f \cdot x|$ from the sum, and change δ to $\delta - |r_k^1|$). We have:

$$\left| r_k^1 + r_k^f \cdot x \right| = \begin{cases} r_k^1 + r_k^f \cdot x & \text{if } x \geq -\frac{r_k^1}{r_k^f} \text{ and } r_k^f > 0 \\ -\left(r_k^1 + r_k^f \cdot x \right) & \text{if } x \leq -\frac{r_k^1}{r_k^f} \text{ and } r_k^f > 0 \\ r_k^1 + r_k^f \cdot x & \text{if } x \leq -\frac{r_k^1}{r_k^f} \text{ and } r_k^f < 0 \\ -\left(r_k^1 + r_k^f \cdot x \right) & \text{if } x \geq -\frac{r_k^1}{r_k^f} \text{ and } r_k^f < 0 \end{cases} \quad (10)$$

We compute the m values $-\frac{r_k^1}{r_k^f}$ for $1 \leq k \leq n$, and sort these values in increasing order into an array $X^\dagger[1..n]$. We remove all values that are < 0 or > 1 , add the values 0 and 1, and remove all duplicates to get the sorted array $X[1..n']$ with no duplicates, and in increasing order, with $X[1] = 0$, and $X[n'] = 1$. Observe that for $X[k] \leq x \leq X[k+1]$, each of $|r_k^1 + r_k^f \cdot x|$ is either $r_k^1 + r_k^f \cdot x$, or $-\left(r_k^1 + r_k^f \cdot x \right)$, *i.e.*, the value of $r_k^1 + r_k^f \cdot x$ is either ≥ 0 throughout, or ≤ 0 throughout the interval. Using this fact, we determine the form of the function $\sum_{k=1}^n |r_k^1 + r_k^f \cdot x|$ over the interval $X[j] \leq x \leq X[j+1]$ as follows. We have:

$$\sum_{k=1}^n |r_k^1 + r_k^f \cdot x| = \sum_{k=1}^n (-1)^{\mu_j(k)} \cdot \left(r_k^1 + r_k^f \cdot x \right) \text{ over the interval } X[j] \leq x \leq X[j+1] \text{ where,} \quad (11)$$

$$\mu_j(k) = \begin{cases} 0 & \text{if } X[j] \geq -\frac{r_k^1}{r_k^f} \text{ and } r_k^f > 0 \\ 1 & \text{if } X[j+1] \leq -\frac{r_k^1}{r_k^f} \text{ and } r_k^f > 0 \\ 0 & \text{if } X[j+1] \leq -\frac{r_k^1}{r_k^f} \text{ and } r_k^f < 0 \\ 1 & \text{if } X[j] \geq -\frac{r_k^1}{r_k^f} \text{ and } r_k^f < 0 \end{cases} \quad (12)$$

We prove that the function $\mu()$ above is well defined.

1. First, we show that either $X[j] \geq -\frac{r_k^1}{r_k^f}$, or $X[j+1] \leq -\frac{r_k^1}{r_k^f}$ hold for all $1 \leq k, j \leq n'$. This is because if this condition does not hold, then both $X[j] < -\frac{r_k^1}{r_k^f}$ and $X[j+1] > -\frac{r_k^1}{r_k^f}$ hold, *i.e.*, $X[j] < -\frac{r_k^1}{r_k^f} < X[j+1]$, which is impossible since X is a sorted array which contains $-\frac{r_k^1}{r_k^f}$.
2. We show both $X[j] \geq -\frac{r_k^1}{r_k^f}$ and $X[j+1] \leq -\frac{r_k^1}{r_k^f}$ cannot hold. If both these hold, then $X[j] \geq -\frac{r_k^1}{r_k^f} \geq X[j+1]$; which is impossible since X is an sorted array in increasing order with no duplicates.

The fact that $\left| r_k^1 + r_k^f \cdot x \right|$ over the interval $X[j] \leq x \leq X[j+1]$ equals $(-1)^{\mu_j(k)} \cdot \left(r_k^1 + r_k^f \cdot x \right)$ then follows from Equation 10. Thus, Equation 11 is correct.

Once the function $\mu_j()$ has been computed, the function $\sum_{k=1}^n \left| r_k^1 + r_k^f \cdot x \right|$ can be written as $\left(\sum_{k=1}^n (-1)^{\mu_j(k)} \cdot r_k^1 \right) + \left(\sum_{k=1}^n (-1)^{\mu_j(k)} \cdot r_k^f \right) \cdot x$ over the interval $X[j] \leq x \leq X[j+1]$. This function is either monotonically non-decreasing, or non-increasing. The maximum value of $\sum_{k=1}^n \left| r_k^1 + r_k^f \cdot x \right|$ over the interval $X[j] \leq x \leq X[j+1]$ is thus,

$$\begin{cases} \sum_{k=1}^n (-1)^{\mu_j(k)} \cdot \left(r_k^1 + r_k^f \cdot X[j+1] \right) & \text{if } \left(\sum_{k=1}^n (-1)^{\mu_j(k)} \cdot r_k^f \right) \geq 0 \\ \sum_{k=1}^n (-1)^{\mu_j(k)} \cdot \left(r_k^1 + r_k^f \cdot X[j] \right) & \text{if } \left(\sum_{k=1}^n (-1)^{\mu_j(k)} \cdot r_k^f \right) < 0 \end{cases}$$

The maximum value of $\sum_{k=1}^n \left| r_k^1 + r_k^f \cdot x \right|$ over the interval $0 \leq x \leq 1$ can then be obtained as the maximum of the maximum values of $\sum_{k=1}^n \left| r_k^1 + r_k^f \cdot x \right|$ over the n' intervals $X[j] \leq x \leq X[j+1]$ for $1 \leq j < n'$.

Computing and sorting the $-r_k^1/r_k^f$ values takes $O(n \cdot \log(n))$ time. Computing the function $\mu_j()$ takes time $O(n)$ for each j , thus computing all the functions $\mu_j()$ takes time $O(n^2)$. This also means that computing the maximums for all of the n' intervals takes $O(n^2)$ time. The rest of the steps take time $O(n)$. Putting everything together, we have the following proposition.

Proposition 10. *Given polygonal curves f and g with values in \mathbb{R}^n , the intersection of the free space $\text{Free}_\delta(f, g)$ with the boundary of the cell i, j can be computed in time $O(n^2)$ for the L_1 norm.* \square

Computing Free Space Cell Boundaries for L_2 . We show how to compute the free space boundary at the cell boundary $0 \leq \rho_f^* \leq 1; \rho_g^* = 1$, where $(\rho_f^*, \rho_g^*) = (\rho_f - i, \rho_g - j)$. The computation for the other boundaries is similar. Substituting $\rho_g^* = 1$ in Equation 8, denoting ρ_f^* as

x , and simplifying, we get:

$$\left(\sum_{k=1}^n (d_k^f)^2\right) \cdot x^2 + \left(\left(\sum_{k=1}^n 2 \cdot d_k^f \cdot d_k^g\right) + \left(\sum_{k=1}^n 2 \cdot d_k^0 \cdot d_k^f\right)\right) \cdot x + \sum_{k=1}^n \left((d_k^g)^2 + 2 \cdot d_k^0 \cdot d_k^g + (d_k^0)^2\right) - \delta^2 \leq 0.$$

This is of the form

$$A \cdot x^2 + B \cdot x + C \leq 0$$

We wish to find the maximum and minimum values of $x \in [0, 1]$ which satisfy the above equation. The equation $A \cdot x^2 + B \cdot x + C = 0$ has two roots given by $\frac{-b \pm \sqrt{B^2 - 4AC}}{2A}$. The following three cases arise.

- $B^2 - 4A \cdot C < 0$, thus, there are no real roots to $A \cdot x^2 + B \cdot x + C = 0$. This means that for all x , the value of $A \cdot x^2 + B \cdot x + C$ is either always greater than 0, or always less than 0. If $C > 0$, then $A \cdot x^2 + B \cdot x + C$ is always greater than 0 (using $x = 0$), thus, the free space boundary is empty. If $C < 0$, then $A \cdot x^2 + B \cdot x + C$ is always less than 0, thus, the free space boundary is the entire segment $0 \leq x \leq 1$ at $y = 1$. Note that we cannot have $C = 0$ as $x = 0$ is not a solution to $A \cdot x^2 + B \cdot x + C = 0$.
- $B^2 - 4A \cdot C = 0$; thus there is exactly one real root x^\dagger to $A \cdot x^2 + B \cdot x + C = 0$. This means that for all $x \neq x^\dagger$, the value of $A \cdot x^2 + B \cdot x + C$ is either always greater than 0, or always less than 0.
 - If $C \leq 0$, then $A \cdot x^2 + B \cdot x + C$ is always less than 0 for $x \neq x^\dagger$, and at $x = x^\dagger$ the value is 0. Hence, the free space boundary is the entire segment $0 \leq x \leq 1$ at $y = 1$.
 - If $C > 0$, then $A \cdot x^2 + B \cdot x + C$ is always greater than 0 for $x \neq x^\dagger$. Thus, the only x such that $A \cdot x^2 + B \cdot x + C \leq 0$ is $x = x^\dagger$. Hence the free space boundary is the singleton point x^\dagger if $x^\dagger \in [0, 1]$; otherwise it is the emptyset.
- $B^2 - 4A \cdot C > 0$; which means that there are two real roots to $A \cdot x^2 + B \cdot x + C = 0$. The following two situations can arise Let the two roots of $A \cdot x^2 + B \cdot x + C \leq 0$ be x_1^\dagger and x_2^\dagger with

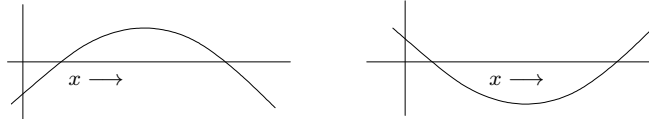


Figure 5: Graphs of $A \cdot x^2 + B \cdot x + C$.

$x_1^\dagger < x_2^\dagger$. The derivative at $A \cdot x^2 + B \cdot x + C$ is $2 \cdot A \cdot x + B$, thus we have the first situation if $2 \cdot A \cdot x_1^\dagger + B > 0$, and the second situation otherwise.

- If $2 \cdot A \cdot x_1^\dagger + B > 0$, then we have the first situation and hence the free space boundary is $[0, x_1^\dagger] \cup [x_2^\dagger, 1]$.
- If $2 \cdot A \cdot x_1^\dagger + B < 0$, then we have the second situation and hence the free space boundary is $[0, 1] \cap [x_1^\dagger, x_2^\dagger]$.

We note that the preceeding computation steps take $O(n)$ time in total.

Computing Free Space Cell Boundaries for L_∞ . We show how to compute the free space boundary at the cell boundary $0 \leq \rho_f^* \leq 1; \rho_g^* = 1$, where $(\rho_f^*, \rho_g^*) = (\rho_f - i, \rho_g - j)$. The computation for the other boundaries is similar. We use Proposition 5. We have that the free space boundary

at $0 \leq \rho_f^* \leq 1; \rho_g^* = 1$ for the k -th curve components f_k and g_k is the set of (ρ_f^*, ρ_g^*) with $\rho_g^* = 1$ such that $|f_k(\rho_f^*) - g_k(\rho_g^*)| \leq \delta$. Using Equation 3, and substituting $\rho_g^* = 1$, we get that

$$|f_k(i) - g_k(j+1) + (f(i+1) - f(i)) \cdot \rho_f^*| \leq \delta$$

If $f(i+1) = f(i)$, then the above equation gives all ρ_f^* as being valid if $|f_k(i) - g_k(j+1)| \leq \delta$ (thus the free space boundary if the entire segment $0 \leq \rho_f^* \leq 1; \rho_g^* = 1$); and no ρ_f^* as being valid otherwise (thus the free space boundary is the emptyset).

If $f(i+1) \neq f(i)$, then the equation for the k -th components states:

$$\rho_f^* \in \begin{cases} [\frac{-(f_k(i)-g_k(j+1))-\delta}{f(i+1)-f(i)}, \frac{-(f_k(i)-g_k(j+1))+\delta}{f(i+1)-f(i)}] & \text{if } f(i+1) - f(i) > 0 \\ [\frac{-(f_k(i)-g_k(j+1))+\delta}{f(i+1)-f(i)}, \frac{-(f_k(i)-g_k(j+1))-\delta}{f(i+1)-f(i)}] & \text{if } f(i+1) - f(i) < 0 \end{cases}$$

Thus, in all cases, the k -th component equations give us the valid ρ_f^* as being the set $[x_k, x'_k]$ with x_k, x'_k determined as above. Using Proposition 5, we get that the free space boundary at $0 \leq \rho_f^* \leq 1; \rho_g^* = 1$ is the emptyset if $[x_k, x'_k] \cap [0, 1] = \emptyset$ for some k , otherwise, it is $[x_f, x'_f] = [0, 1] \cap \bigcap_{k=1}^n [x_k, x'_k]$. This intersection can be determined as follows: $x_f = \max(0, \max\{x_k\})$, and $x'_f = \min(1, \min\{x'_k\})$. We note that the preceeding computation steps take $O(n)$ time in total.

Computing Free Space Cell Boundaries for L_1^S, L_2^S, L_∞^S . Let $\langle f, f_{n+1} \rangle$ be a curve from $[a_f, b_f]$ to $\mathbb{R}^n \times \mathbb{R}$, and similarly for $\langle g, g_{n+1} \rangle$ as in Proposition 9. To compute the free space boundaries of these two curves for L_1^S, L_2^S or L_∞^S norms, we first compute the free space boundary for the norm L_1, L_2 or L_∞ for the curves f, g . Then we compute the free space corresponding to the last component, f_{n+1}, g_{n+1} – this computation is the same as the computation of the free space for individual coordinate components for L_∞ . Then we intersect the two boundaries. The time taken is $O(n^2)$ for L_1^S , and $O(n)$ for L_2^S and L_∞^S .

4.4 Algorithm for the Fréchet-Distance Decision Problem Given a Fixed δ

In this section we solve for the Fréchet distance decision problem between two polygonal curves for a given fixed δ . The decision problem is solved with a dynamic programming algorithm on the free space diagram. As noted before, our formulation of the Fréchet distance requires the reparametrizations to be strictly increasing, as opposed to the formulation of [AG95] which only requires non-decreasing reparametrizations. This introduces some complications which we address in our solution.

Consider cell i, j in the free space diagram of two polygonal curves f, g . The cell together with the free space inside it is depicted in Figure 6². The non-shaded portion is the free space. Let $\mathbf{e}_{i,j}^0, \mathbf{e}_{i,j}^1, \mathbf{e}_{i,j}^2$, and $\mathbf{e}_{i,j}^3$ denote the bottom, right, top, and left edges of cell i, j respectively. Thus, $\mathbf{e}_{i,j}^0$ is the edge $[i, i+1] \times \{j\}$, and the other edges are the ones encountered moving anti-clockwise. Let $\mathbf{a}_{i,j}^0$ and $\mathbf{b}_{i,j}^0$ be the starting and ending points of edge $\mathbf{e}_{i,j}^0$, and similarly for the other edges (see Figure 4.4). Given a point $s = (p, q) \in \mathbb{R}^2$, let $\text{first}(s) = p$ denote the first coordinate, and $\text{second}(s) = q$ denote the second coordinate. The points $\mathbf{a}_{i,j}^q, \mathbf{b}_{i,j}^q$ for $0 \leq q \leq 3$ can be obtained for each cell using the results of the previous section for the L_1, L_2, L_∞ and L_1^S, L_2^S, L_∞^S norms. Note that $(\mathbf{a}_{i+1,j}^3, \mathbf{b}_{i+1,j}^3) = (\mathbf{a}_{i,j}^1, \mathbf{b}_{i,j}^1)$; and $(\mathbf{a}_{i,j+1}^0, \mathbf{b}_{i,j+1}^0) = (\mathbf{a}_{i,j+1}^2, \mathbf{b}_{i,j+1}^2)$

²We use the convention that the first coordinate i of cell i, j increases in the horizontal direction, and that the second coordinate in the vertical direction, thus, the same as for the cartesian plane.

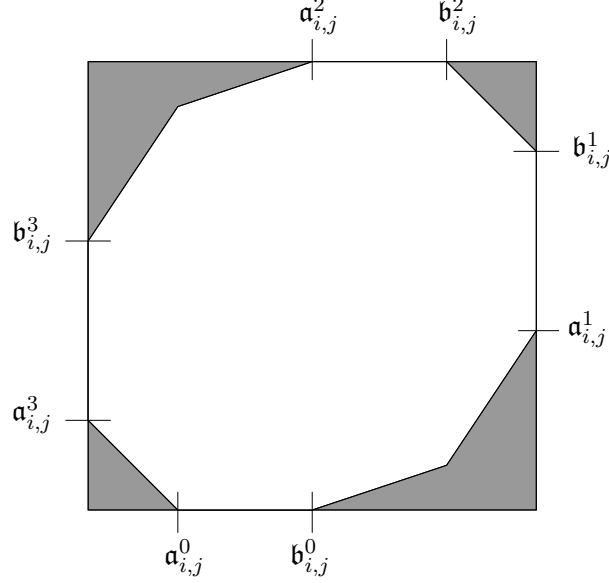


Figure 6: Cell i, j in $\text{Free}_\delta(f, g)$.

In order to utilize Proposition 2 in order to check for the existence of reparametrizations which demonstrate that $\mathcal{F}(f, g) \leq \delta$, we present a dynamic programming based method (based on the sketch in [AG95]) to determine the set of points in $\text{Free}_\delta(f, g)$ that are reachable from $(0, 0)$ by a monotone curve which is strictly increasing in both coordinates. The method iteratively computes the monotone curve reachable set at the 4 cell boundaries. At each cell boundary, this set consists of a closed interval of points, together with possibly one corner point. We define the parameters $\ddot{a}_{i,j}^q, \ddot{b}_{i,j}^q$ for $0 \leq q \leq 3$, and the set $\text{cpoint}_{i,j}$ for cell i, j . The set $\text{cpoint}_{i,j}$ is either empty, or contains the point (i, j) . The ranges of the parameters $\ddot{a}_{i,j}^q$ and $\ddot{b}_{i,j}^q$ are:

- $\left(([i, i+1] \times \{\cdot, +\}) \times \{j, j+1\} \right) \cup \{\perp\}$ for $\ddot{a}_{i,j}^0, \ddot{b}_{i,j}^0$, and $\ddot{a}_{i,j}^2, \ddot{b}_{i,j}^2$. A second coordinate value of j corresponds to the bottom cell boundary, and a value of $j+1$ corresponds to the top cell boundary.
- $\left(\{i, i+1\} \times ([j, j+1] \times \{\cdot, +\}) \right) \cup \{\perp\}$ for $\ddot{a}_{i,j}^1, \ddot{b}_{i,j}^1$, and $\ddot{a}_{i,j}^3, \ddot{b}_{i,j}^3$. A first coordinate value of i corresponds to the left cell boundary, and a value of $i+1$ corresponds to the right cell boundary.

A value $(\langle x, + \rangle, y)$ denotes a point which has the first coordinate value that is ϵ greater than x , for ϵ arbitrarily small, and a second coordinate value that is exactly equal to y . We need this ϵ -formalism as we are only interested in monotone curves in the free space which are *strictly* increasing in both coordinates. The line from $\ddot{a}_{i,j}^q$ to $\ddot{b}_{i,j}^q$ is the subportion of the line from $a_{i,j}^q$ to $b_{i,j}^q$ that is reachable from a monotone curve from $(0, 0)$ (the reachable sub-portion may be open). The special value \perp is used to indicate that the monotone curve reachable portion of cell boundary is empty. The corner point needs to be treated differently for technical reasons. The dynamic programming algorithm based on the sketch from [AG95] for computing the parameters is presented in Algorithm 1³. The requirement of reparametrizations being strictly increasing, instead of only being non-decreasing as

³Note that Algorithm 1 computes whether there are reparametrizations which *achieve* the Fréchet distance δ . This is not the same as determining whether the Fréchet distance is at most δ .

in [AG95] introduces some complications, *e.g.*, now the strictly increasing monotone curve portion at a cell boundary need not be a line segment, rather, it is a line segment together with possibly a corner point. The procedure for computing the $\ddot{\mathbf{a}}_{i,j}^q, \ddot{\mathbf{b}}_{i,j}^q$ values used in the algorithm is given in the appendix.

```

Input: Polygonal curves  $f : [0, m_f] \rightarrow \mathbb{R}^n$  and  $g : [0, m_g] \rightarrow \mathbb{R}^n$ ; and  $\delta \geq 0$ 
foreach  $0 \leq i \leq m_f$  and  $0 \leq j \leq m_g$  do compute  $\mathbf{a}_{i,j}^3, \mathbf{b}_{i,j}^3, \mathbf{a}_{i,j}^0, \mathbf{b}_{i,j}^0$ ;
foreach  $0 \leq i \leq m_f$  do compute  $\ddot{\mathbf{a}}_{i,0}^1, \ddot{\mathbf{b}}_{i,0}^1, \ddot{\mathbf{a}}_{i,0}^3, \ddot{\mathbf{b}}_{i,0}^3$  and  $\text{cpoint}_{i,0}$ ;
                                     /* Vertical reach boundaries of bottom row */
foreach  $0 \leq j \leq m_g$  do compute  $\ddot{\mathbf{a}}_{0,j}^2, \ddot{\mathbf{b}}_{0,j}^2, \ddot{\mathbf{a}}_{0,j}^0, \ddot{\mathbf{b}}_{0,j}^0$  and  $\text{cpoint}_{0,j}$ ;
                                     /* Horizontal reach boundaries of first column */

foreach  $1 \leq i \leq m_f$  do
    foreach  $1 \leq j \leq m_g$  do
        compute  $\text{cpoint}_{i,j}$  from  $\ddot{\mathbf{a}}_{i-1,j-1}^1, \ddot{\mathbf{b}}_{i-1,j-1}^1$  and  $\ddot{\mathbf{a}}_{i-1,j-1}^2, \ddot{\mathbf{b}}_{i-1,j-1}^2$ ;
        compute  $\ddot{\mathbf{a}}_{i,j}^1, \ddot{\mathbf{b}}_{i,j}^1$  from  $\text{cpoint}_{i-1,j-1}$  and  $\ddot{\mathbf{a}}_{i-1,j}^1, \ddot{\mathbf{b}}_{i-1,j}^1$  and  $\ddot{\mathbf{a}}_{i,j-1}^2, \ddot{\mathbf{b}}_{i,j-1}^2$ ;
        compute  $\ddot{\mathbf{a}}_{i,j}^2, \ddot{\mathbf{b}}_{i,j}^2$  from  $\text{cpoint}_{i-1,j-1}$  and  $\ddot{\mathbf{a}}_{i,j-1}^2, \ddot{\mathbf{b}}_{i,j-1}^2$  and  $\ddot{\mathbf{a}}_{i-1,j}^3, \ddot{\mathbf{b}}_{i-1,j}^3$ ;
    end
end
check if  $(m_f, m_g) = \ddot{\mathbf{b}}_{m_f, m_g}^2$ ;

```

Algorithm 1: Dynamic programming algorithm for checking reachability of (m_f, m_g) by a monotone curve

Lemma 2. Let $f : [0, m_f] \rightarrow \mathbb{R}^n$ and $g : [0, m_g] \rightarrow \mathbb{R}^n$ be polygonal curves. There exists a strictly increasing monotone curve from $(0, 0)$ to (m_f, m_g) in the free space diagram of f, g iff the point (m_f, m_g) belongs to the reachable portion of the top cell boundary of cell $(m_f - 1, m_g - 1)$, i.e. $\ddot{\mathbf{b}}_{m_f, m_g}^2 = (m_f, m_g)$. \square

Note that since require reparametrizations to be *strictly* increasing, it might be the case that there is no strictly increasing monotone curve from $(0, 0)$ to (m_f, m_g) in $\text{Free}_\delta(f, g)$, and that such curves exist in $\text{Free}_{\delta+\epsilon'}(f, g)$ for every $\epsilon' > 0$. This presents a complication in determining whether $\mathcal{F}(f, g) \leq \delta$, as a value of δ might be the limit obtained by a sequence of reparametrizations. In the free space formulation, this situation can arise when there is only a horizontal line (*i.e.* a non-decreasing monotone curve) which can cross a cell boundary for δ ; with strictly increasing monotone curves only being available for $\delta' > \delta$. See Figure 7 which depicts this situation for cell i, j in the two free space diagrams, with $\delta' > \delta$.

The non-bijective Fréchet distance. Consider a variant of the Fréchet distance in which we drop the requirement of the reparametrizations α_f, α_g to be *strictly* increasing in Definition 3; and instead only require them to be continuous and non-decreasing. This implies that an entire segment of the curve f can be matched to a single point of g (and vice versa). In the free space approach, it means that we can now consider continuous and non-decreasing curves from $(0, 0)$ to (m_f, m_g) . Let us denote this version of the distance as the non-bijective Fréchet distance ⁴, and

⁴We use the superscript ^{nbj} for entities relating to the non-bijective Fréchet distance.

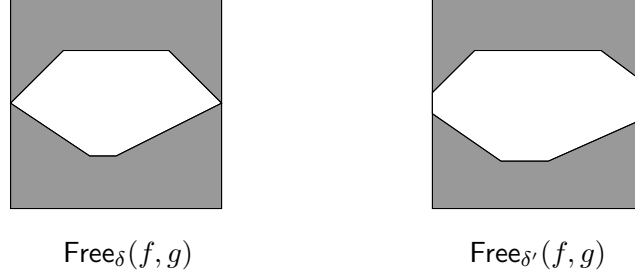


Figure 7: Cell i, j in $\text{Free}_\delta(f, g)$ and $\text{Free}_{\delta'}(f, g)$; with $\delta' > \delta$.

the strictly increasing version as the bijective one. For this version, we cannot have that there exist non-decreasing monotone reparametrizations for every $\delta' > \delta^*$, but not for δ^* . This is because the free space at the cell boundaries is always a closed interval for every δ . We can show that the endpoints of the intersections of these intervals for opposite cell boundaries are continuous functions for $L_1, L_2, L_\infty, L_1^S, L_2^S, L_\infty^S$ norms. Finally, using compactness of $[\delta^*, \delta']$ and continuity, we can show that the intersection of the opposite cell boundaries will be non-empty for δ^* . Thus, for the non-bijective variant of the Fréchet distance, there exist reparametrizations which achieve the Fréchet distance. This gives us the following result.

Proposition 11 ([AG95]). *Let $f : [0, m_f] \rightarrow \mathbb{R}^n$ and $g : [0, m_g] \rightarrow \mathbb{R}^n$ be polygonal curves. There exists a non-decreasing monotone curve from $(0, 0)$ to (m_f, m_g) in the free space diagram $\text{Free}_\delta(f, g)$ iff $\mathcal{F}^{\text{nbij}}(f, g) \leq \delta$.* \square

The algorithm of the bijective Fréchet distance decision problem can be easily modified for the non-bijective version. The next lemma shows that the distance under the two semantics remains the same.

Lemma 3. *Let $f : [a_f, b_f] \rightarrow \mathcal{V}$ and $g : [a_g, b_g] \rightarrow \mathcal{V}$ be curves such that $a_f \neq b_f$ and $a_g \neq b_g$. We have $\mathcal{F}^{\text{nbij}}(f, g) = \mathcal{F}(f, g)$.*

Proof. Suppose $\mathcal{F}^{\text{nbij}}(f, g) \leq \delta$. We show $\mathcal{F}(f, g) \leq \delta$ (the other direction is trivial). Since $\mathcal{F}^{\text{nbij}}(f, g) \leq \delta$, given any $\epsilon > 0$, there exist reparametrizations $\alpha_f^{\text{nbij}}, \alpha_g^{\text{nbij}}$ as in Definition 3 (for non-bijective Fréchet distance) that are continuous and non-decreasing such that $\max_{0 \leq \theta \leq 1} \|f(\alpha_f^{\text{nbij}}(\theta)) - g(\alpha_g^{\text{nbij}}(\theta))\| \leq \delta + \epsilon$. Since f and g are continuous, and each is defined over an interval that is not a singleton point, given any $\epsilon' > 0$, we can obtain parameterizations α_f, α_g from $\alpha_f^{\text{nbij}}, \alpha_g^{\text{nbij}}$ which are continuous and *strictly* increasing such that $\max_{0 \leq \theta \leq 1} \|f(\alpha_f(\theta)) - g(\alpha_g(\theta))\| \leq \delta + \epsilon + \epsilon'$. The result follows choosing $\epsilon, \epsilon' \rightarrow 0$. \square

The parameters $\ddot{\mathbf{a}}_{i,j}^q, \ddot{\mathbf{b}}_{i,j}^q$ for the non-bijective Fréchet distance variant can be computed inductively as before. This, together with the results of the present section gives us the following result.

Proposition 12 (The Fréchet distance decision problem). *Let $f : [0, m_f] \rightarrow \mathbb{R}^n$ and $g : [0, m_g] \rightarrow \mathbb{R}^n$ be polygonal curves, and let \mathbb{R}^n be equipped with the norm χ . Given $\delta \geq 0$, there is an algorithm running in $O(m_f \cdot m_g \cdot H(\chi))$ time which decides whether $\mathcal{F}(f, g) \leq \delta$, where $H(\chi)$ is the time required to determine the parameters $\mathbf{a}_{i,j}^q, \mathbf{b}_{i,j}^q$ for $0 \leq q \leq 3$ for a cell i, j (i.e. to compute the*

free space boundaries for two lines), for the norm χ . The algorithm also decides whether there exist reparametrizations α_f, α_g such that $\max_{0 \leq \theta \leq 1} \|f(\alpha_f(\theta)) - g(\alpha_g(\theta))\| \leq \delta$.

Proof. The existence of the algorithm, and its complexity follows from Lemmas 2, 3, and the dynamic programming algorithm for computing the parameters on free space boundaries of cells.

For the second objective of the algorithm, we first determine whether $\mathcal{F}^{\text{nbij}}(f, g) \leq \delta$, and if so check whether there exists a strictly increasing monotone curve from $(0, 0)$ to (m_f, m_g) in $\text{Free}_\delta(f, g)$. If there exists such a curve, then the optimal Fréchet distance can be achieved using strictly increasing reparametrizations α_f, α_g such that $\max_{0 \leq \theta \leq 1} \|f(\alpha_f(\theta)) - g(\alpha_g(\theta))\| \leq \delta$. If there does not exist such a curve, then the value δ cannot be achieved by any strictly increasing reparametrizations α_f, α_g . □

Limiting the Reparametrizations using Windows. Given polygonal curves $f : [0, m_f] \rightarrow \mathbb{R}^n$ and $g : [0, m_g] \rightarrow \mathbb{R}^n$, the Fréchet reparametrizations allow portions of the line segment $f_{[j]}$ to be matched to line segments $g_{[k]}$ for any $k \geq 0$. An additional window requirement on the reparametrizations can be imposed which requires that the curve segment $f_{[k]}$ be only be allowed to match the curves segments $g_{[\max(0, k-W)], g_{[\max(0, k-W+1)], \dots, g_{[\min(k+W, m_g)]}}$, thus a window of W around each index, and $|\alpha_f(\nu) - \alpha_g(\nu)| \leq W$ for all $\nu \in [0, 1]$. This means that only the free cells i, j such that $|i - j| \leq W$ are relevant. There are at most $W \cdot \max(m_f, m_g)$ such cells. The rationale behind the window requirement is that for practical applications, we are often only interested in reparametrizations for which the maximal curve parameter deviation is bounded by a constant. The window requirement can be used to bring down the complexity of the Fréchet distance decision problem to $O(W \cdot \max(m_f, m_g) \cdot H(\chi))$, where $H(\chi)$ is as in Proposition 12.

4.5 Algorithm for Determining the Value of the Fréchet Distance

In this section we use the non-bijective Fréchet distance formulation.

Critical Values of δ . The free space $\text{Free}_\delta^{\text{nbij}}(f, g)$ keeps increasing as we increase δ . As the free space gets bigger, new paths open up which make a non-decreasing curve from $(0, 0)$ to (m_f, m_g) in the free space feasible. Consider the free space diagram. We explore when new paths become feasible. Based on the sketch in [AG95] for \mathbb{R}^2 , this happens at the following values of δ for \mathbb{R}^n .

1. Values of δ for which $(0, 0)$ and (m_f, m_g) “get” into the free space. These two values of δ are $\|f(0) - g(0)\|$ and $\|f(m_f) - g(m_g)\|$.
2. Values of δ which enable a monotone non-decreasing curve to enter a cell. A curve can enter a cell i, j if either $\mathbf{b}_{i,j}^3$ or $\mathbf{b}_{i,j}^0$ is not equal to \perp (the case where a curve enters cell i, j through the corner point (i, j) is covered by this condition). Let δ be the least value which makes $\mathbf{b}_{i,j}^3 = (i, \Delta_j) \neq \perp$. This is the least value of δ for which we have $\text{Line}(\mathbf{b}_{i,j}^3, \mathbf{a}_{i,j}^3)$ is non-empty. Thus, this is the least value of δ for which the point $f(i)$ is at most δ away from the line segment $g_{[j]}$. It follows that this value of δ is just the distance of the point $f(i)$ from the line segment $g_{[j]}$. Similarly, the least value of δ which makes $\mathbf{b}_{i,j}^0 \neq \perp$ is the distance of the point $g(j)$ from the line segment $f_{[i]}$. Since there are $m_f \cdot m_g$ cells, there are $2 \cdot m_f \cdot m_g$ such critical δ values.
3. Value of δ which enables a curve to go from cell i, j to cell k, j (for $k > i$), that is a value which makes the free space big enough so that at least a horizontal line can (possibly) go

from cell i, j to cell k, j . This happens when $\text{second}(\mathbf{a}_{i,j}^1) \leq \text{second}(\mathbf{b}_{k,j}^3)$. Observe that if $\text{second}(\mathbf{a}_{i,j}^1) > \text{second}(\mathbf{b}_{k,j}^3)$, then even if the rest of the cells are fully free, there cannot be a curve from cell i, j to cell k, j . When $\text{second}(\mathbf{a}_{i,j}^1)$ becomes equal to, or greater than $\text{second}(\mathbf{b}_{k,j}^3)$, by increasing δ , it enables a curve to go from cell i, j to cell k, j . We can obtain the value of this special δ as follows. For a point s , let $\text{Ball}(s, \delta)$ denote the set of points which are at most δ away from s , formally $\text{Ball}(s, \delta) = \{q \mid \|q - s\| \leq \delta\}$. We are interested in the least δ such that there is some point on the line segment $g_{[j]}$ that is at most δ away from the point $f(i)$, and also from the point $f(k)$. Mathematically, this is equivalent to finding the least δ such that $\text{Ball}(f(i), \delta) \cap \text{Ball}(f(k), \delta) \cap g_{[j]}$ is non-empty. We prove in the next lemma that such a least δ exists. These δ values are called *horizontally clamped δ values*.

See Figure 8 for a horizontally clamped situation, where $\mathbf{a}_{i,j}^1 = (i+1, \Delta)$, and $\mathbf{b}_{k,j}^3 = (k, \Delta)$, i.e., both points have the same ρ_g values. Note that the only monotone non-decreasing curve which can pass from cell i, j to cell k, j must have a horizontal straight line segment from point $\mathbf{a}_{i,j}^1$ to point $\mathbf{b}_{k,j}^3$.



Figure 8: Horizontally clamped cells (i, j) and (k, j) in $\text{Free}_\delta(f, g)$.

For each $0 \leq i \leq m_f - 1$ and $0 \leq j \leq m_g - 1$, there are $m_f - 1 - i$ such critical δ values for cell i, j . Thus, for each j , there are $\sum_{i=0}^{m_f-1} (m_f - 1 - i) = (m_f - 1) \cdot (m_f - 2) / 2$ such critical δ values. Hence overall there are $(m_g - 1) \cdot (m_f - 1) \cdot (m_f - 2) / 2$ such values.

A similar analysis applies when we consider vertical lines, and in this case there are $(m_f - 1) \cdot (m_g - 1) \cdot (m_g - 2) / 2$ such critical *vertically clamped δ values*.

Lemma 4. Let s_1, s_2, l_1, l_2 be four points in the space \mathbb{R}^n with the norm $L_1, L_2, L_\infty, L_1^S, L_2^S$, or L_∞^S . There exists δ^* such that

$$\delta^* = \min_{\delta \geq 0} \{ \delta \mid \text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(l_1, l_2) \neq \emptyset \}.$$

Proof. Consider any norm under consideration. The equation of the line is $l_1 + \lambda \cdot (l_2 - l_1)$ for $0 \leq \lambda \leq 1$. Define the function $h(\lambda) = \max(\|l_1 + \lambda \cdot (l_2 - l_1) - s_1\|, \|l_1 + \lambda \cdot (l_2 - l_1) - s_2\|)$. In words, λ gives a point on the line, and $h(\lambda)$ is the maximum of the distances to points s_1 and s_2 . Observe that

$$\inf_{\delta \geq 0} \{ \delta \mid \text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(l_1, l_2) \neq \emptyset \} = \inf \{ \delta \in h([0, 1]) \}.$$

It can be shown that h is continuous (and bounded) on $[0, 1]$. Since $[0, 1]$ is compact, and h is continuous, we have that $h([0, 1])$ is compact, and thus closed (and bounded). Thus, $h([0, 1])$ contains the infimum $\inf h([0, 1])$. Thus, there is a point on the line $\text{Line}(l_1, l_2)$ which is at most δ^* away from s_1 , and from s_2 . This proves the statement of the lemma. \square

To compute the least possible value of δ which makes a non-decreasing curve from $(0, 0)$ to (m_f, m_g) in the free space $\text{Free}_\delta(f, g)$ feasible, we find the least critical value amongst the $O(m_g \cdot m_f^2 + m_f \cdot m_g^2)$ values of δ for which there is such a curve. To do this, we sort the $O(m_g \cdot m_f^2 + m_f \cdot m_g^2)$ values and perform a binary search using the decision procedure from the previous section.

Theorem 1 (Computing the Fréchet distance). *Let $f : [0, m_f] \rightarrow \mathbb{R}^n$ and $g : [0, m_g] \rightarrow \mathbb{R}^n$ be polygonal curves, and let \mathbb{R}^n be equipped with the norm χ . There is an algorithm running in time*

$$O\left((m_g \cdot m_f^2 + m_f \cdot m_g^2) \cdot (P(\chi) + \log(m_g \cdot m_f)) + m_f \cdot m_g \cdot H(\chi)\right)$$

which computes the value $\mathcal{F}(f, g)$ where (i) $H(\chi)$ is the time required to determine the parameters $\mathbf{a}_{i,j}^q, \mathbf{b}_{i,j}^q$ for $0 \leq q \leq 3$ for a cell i, j (i.e. to compute the free space boundaries for two lines), and (ii) $P(\chi)$ is the time required to compute a critical value of δ as outlined previously for the desired norm χ . \square

The Fréchet Distance with Windows. As for the decision problem, we can apply a window of W in the computation of the Fréchet distance. This allows us to restrict our attention to $W \cdot \max(m_f, m_g)$ cells. For each of these cells, there are at most W horizontal and W vertical clamping δ . Thus, there are at most $W^2 \cdot \max(m_f, m_g)$ candidate values for the Fréchet distance. We run a binary search on these, thus, we only have to run the Fréchet distance decision algorithm (with windows) at most $O(\log(W \cdot \max(m_f, m_g)))$ times. Hence the complexity of the entire algorithm with windows is $O\left(W^2 \cdot M \cdot (P(\chi) + \log(W \cdot M)) + W \cdot M \cdot \log(W \cdot M) \cdot H(\chi)\right)$, where $M = \max(m_f, m_g)$; and $H(\chi)$, and $P(\chi)$ are as in Theorem 1. If W can be taken to be a constant, the complexity is $O\left(M \cdot P(\chi) + M \cdot \log(M) \cdot H(\chi)\right)$.

5 Computing the Geometric Primitives

This section is concerned with solving for the following two geometric primitives for the 6 norms $L_1, L_2, L_\infty, L_1^S, L_2^S, L_\infty^S$ in the space \mathbb{R}^n . As shown in the previous section, these primitives can be used to compute a set of “critical” value of δ which contains the Fréchet distance value for polygonal curves. The last 3 norms are required by the Fréchet distance based algorithm of the previous section for computing the Skorokhod distance between two linear interpolation traces.

1. The distance of a point s to a line $\text{Line}(z, z')$.
2. Given four points s_1, s_2, z, z' , the least $\delta \geq 0$ such that $\text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z')$ is non-empty.

We present the solutions for the various norms. The formal definition of the distance from a point to a set is given below.

Definition 7 (Distance). The distance of a point x to a set S in a metric space \mathcal{V} is defined to be $\inf_{y \in S} \mathcal{D}_\mathcal{V}(y, x)$, where $\mathcal{D}_\mathcal{V}$ is the metric associated with the metric space \mathcal{V} .

5.1 L_1 -norm

In this section, all norms are L_1 -norms.

Proposition 13 (Distance of point to line: L_1). *The distance $\mathcal{D}_{L_1}(s, \text{Line}(z, z'))$ from a point $s \in \mathbb{R}^n$ to the affine line segment between two distinct points z and z' in \mathbb{R}^n is the solution of the following linear program with $2 \cdot n + 1$ variables U_i^+, U_i^- for $1 \leq i \leq n$, and λ .*

$$\begin{aligned}
& \text{minimize} && \sum_{i=1}^n (U_i^+ + U_i^-) \\
& \text{subject to} && U_i^+ - U_i^- = z_i - s_i + \lambda \cdot (z'_i - z_i) && \text{for } 1 \leq i \leq n \\
& && 0 \leq \lambda \leq 1 \\
& && U_i^+ \geq 0 && \text{for } 1 \leq i \leq n \\
& && U_i^- \geq 0 && \text{for } 1 \leq i \leq n
\end{aligned}$$

Proof. The equation of the affine line segment between two points z and z' is $z + \lambda v$ with $0 \leq \lambda \leq 1$, where $v = z' - z$. By definition, we have

$$\mathcal{D}_{L_1}(s, \text{Line}(z, z')) = \inf_{0 \leq \lambda \leq 1} \sum_{i=1}^n |z_i - s_i + \lambda \cdot v_i|$$

We transform the above into a standard linear program in two steps. Let $U_i = z_i - s_i + \lambda \cdot v_i$ be n new variables. Then, we have $\mathcal{D}_{L_1}(s, \text{Line}(z, z'))$ to be a solution of the following constraint problem:

$$\begin{aligned}
& \text{minimize} && \sum_{i=1}^n |U_i| \\
& \text{subject to} && U_i = z_i - s_i + \lambda \cdot v_i && \text{for } 1 \leq i \leq n \\
& && 0 \leq \lambda \leq 1
\end{aligned}$$

The absolute values in the objective function can be removed as follows (based on the sketch in [BR06]). Let $U_i = U_i^+ - U_i^-$ such that $U_i^+ \geq 0$ and $U_i^- \geq 0$; and $|U_i| = U_i^+ + U_i^-$. The previous constraint problem has the same solution as the linear program in the statement of the lemma. The proof of correctness of the last transformation can be found in [BR06]. \square

Computation of least δ such that $\text{Ball}(q, \delta) \cap \text{Ball}(q', \delta) \cap \text{Line}(z, z')$ is non-empty.

The least δ can be computed as follows. If $q = q'$, then the least δ is simply $\mathcal{D}_{L_1}(q, \text{Line}(z, z'))$ by definition. If $z = z'$, then the least δ is $\min(\|q - z\|, \|q' - z\|)$.

Consider the remaining case where $q \neq q'$ and $z \neq z'$. The equation of the affine line segment between two points z and z' is $z + \lambda \cdot v$ with $0 \leq \lambda \leq 1$, where $v = z' - z$. We have the following

optimization problem involving two variables δ, t :

$$\begin{aligned}
& \text{minimize } \delta \\
& \text{subject to } \sum_{i=1}^n |q_i - (z_i + \lambda \cdot v_i)| \leq \delta \\
& \quad \sum_{i=1}^n |q'_i - (z_i + \lambda \cdot v'_i)| \leq \delta \\
& \quad 0 \leq \lambda \leq 1 \\
& \quad 0 \leq \delta
\end{aligned}$$

We add $2 \cdot n$ new variables U_i, U'_i for $1 \leq i \leq n$ such that $U_i = q_i - (z_i + \lambda \cdot v_i)$, and $U'_i = q'_i - (z_i + \lambda \cdot v'_i)$. The above optimization problem can then be written as:

$$\begin{aligned}
& \text{minimize } \delta \\
& \text{subject to } \sum_{i=1}^n |U_i| - \delta \leq 0 \\
& \quad \sum_{i=1}^n |U'_i| - \delta \leq 0 \\
& \quad U_i = q_i - (z_i + \lambda \cdot v_i) \quad \text{for } 1 \leq i \leq n \\
& \quad U'_i = q'_i - (z_i + \lambda \cdot v'_i) \quad \text{for } 1 \leq i \leq n \\
& \quad 0 \leq \lambda \leq 1 \\
& \quad 0 \leq \delta
\end{aligned} \tag{13}$$

We remove the absolute values in the constraints as follows. We introduce another $2 \cdot n$ new variables $absU_i, absU'_i$ for $1 \leq i \leq n$ such that $absU_i \geq U_i$ and $absU_i \geq -U_i$ and similarly for

$absU'_i$. Consider the linear program:

$$\begin{aligned}
& \text{minimize } \delta \\
& \text{subject to } \left(\sum_{i=1}^n absU_i \right) - \delta \leq 0 \\
& \quad \left(\sum_{i=1}^n absU'_i \right) - \delta \leq 0 \\
& \quad U_i = q_i - (z_i + \lambda \cdot v_i) \quad \text{for } 1 \leq i \leq n \\
& \quad U'_i = q'_i - (z_i + \lambda \cdot v_i) \quad \text{for } 1 \leq i \leq n \\
& \quad U_i - absU_i \leq 0 \quad \text{for } 1 \leq i \leq n \\
& \quad -U_i - absU_i \leq 0 \quad \text{for } 1 \leq i \leq n \\
& \quad U'_i - absU'_i \leq 0 \quad \text{for } 1 \leq i \leq n \\
& \quad -U'_i - absU'_i \leq 0 \quad \text{for } 1 \leq i \leq n \\
& \quad absU_i \geq 0 \quad \text{for } 1 \leq i \leq n \\
& \quad absU'_i \geq 0 \quad \text{for } 1 \leq i \leq n \\
& \quad 0 \leq \lambda \leq 1 \\
& \quad 0 \leq \delta
\end{aligned} \tag{14}$$

Note that $absU_i \geq U_i$ and $absU_i \geq -U_i$ only ensure $absU_i \geq |U_i|$, equality is *not* guaranteed. However, the new constraint $\sum_{i=1}^n absU_i - \delta \leq 0$ ensures that $absU_i$ can be taken to be $|U_i|$ in the computation of the optimum in the linear program. This can be seen as follows. Let δ^\dagger be the value of the original constraint problem 13. Suppose $\delta = \delta^*$ is the optimum value of the linear program 14. It can be seen that $\delta^* \leq \delta^\dagger$ as the feasible region is bigger in the linear program 14. For this optimum δ^* , suppose we have $absU_k = \alpha$ such that $\alpha > |U_k|$ for some k . Since $absU_i \geq 0$ for all i , we must have $\delta^* > 0$ by the inequality $(\sum_{i=1}^n absU_i) - \delta^* \leq 0$. We have the following two cases.

1. Suppose $(\sum_{i=1}^n absU'_i) - \delta^* < 0$. Consider a new value of $absU_k$ which is equal to $|U_k|$. Observe that all the constraints are still satisfied with this new value of $absU_k$. Also observe that we can decrease the value of δ , from δ^* by a non-zero amount $\min((\delta^* - (\sum_{i=1}^n absU'_i), \alpha - |U_k|)$, and still satisfy the constraints of the linear program 14. This is a contradiction as δ^* was assumed to be the optimal value of the program.
2. Suppose $(\sum_{i=1}^n absU'_i) - \delta^* = 0$ in this instantiation of the variables. Then, if we decrease the value of $absU_k$ from α to $|U_k|$, all the constraints are still satisfied, in particular $(\sum_{i=1}^n absU_i) - \delta^* \leq 0$ still holds. Thus, we can set $absU_k = |U_k|$ without changing the optimal value of the objective function. Iterating over i , we see that we can set $absU_i = |U_i|$ for all i without changing the optimal value of the objective function.

Repeating the argument for the $absU'_i$ variables, we get that $absU_i$ and $absU'_i$ can be taken to be $|U_i|$ and $|U'_i|$ respectively in the computation of the optimum in the linear program 14.

The results of the proceeding discussion are summarized in the following proposition.

Proposition 14 (Computation of least δ such that $\text{Ball}(q, \delta) \cap \text{Ball}(q', \delta) \cap \text{Line}(z, z')$ is non-empty:

L_1). Let q, q', z, z' be points in \mathbb{R}^n . The value of $\inf_{\text{Ball}(q, \delta) \cap \text{Ball}(q', \delta) \cap \text{Line}(z, z') \neq \emptyset} \delta$ for the L_1 norm is:

$$\begin{cases} \mathcal{D}_{L_1}(q, \text{Line}(z, z')) & \text{if } q = q' \\ \min(\|q - z\|, \|q' - z\|) & \text{if } q \neq q' \text{ and } z = z' \\ \text{The value of the linear program 14} & \text{otherwise.} \end{cases}$$

□

5.2 L_2 -norm

In this section, all norms are L_2 -norms. Let \mathbb{R}^n be the n -dimensional vector space over \mathbb{R} with the L_2 -norm. We identify an n -tuple $x = (x_1, \dots, x_n)$ with $x_i \in \mathbb{R}$ for $1 \leq i \leq n$ with the corresponding vector $\vec{x} = (x_1, \dots, x_n)$ in the vector space \mathbb{R}^n . The vector equation of the affine line segment between two points z and z' is $\vec{z} + \lambda \cdot (\vec{z}' - \vec{z})$ with $0 \leq \lambda \leq 1$. Given two vectors \vec{x}, \vec{y} , we denote their dot product by $\vec{x} \odot \vec{y}$. Formally, for $\vec{x} = (x_1, \dots, x_n)$ and $\vec{y} = (y_1, \dots, y_n)$ we have

the *dot product* $\vec{x} \odot \vec{y}$ to be the scalar $\sum_{i=1}^n x_i \cdot y_i$. Given two vectors \vec{x}, \vec{y} , the angle θ between them

is defined by the relation $\cos(\theta) \triangleq \frac{\vec{x} \odot \vec{y}}{\|\vec{x}\| \|\vec{y}\|}$. Two vectors are said to be *perpendicular* if the angle between them is 90 degrees, i.e. if $\vec{x} \odot \vec{y} = 0$. For the basics of L_2 vector geometry, we refer the reader to [Blo79, Slo01]. When we want to emphasize vector operations such as the dot product, we use the vector notation, e.g., \vec{x} .

Proposition 15 (Distance of point to line: L_2). *The distance from a point $s \in \mathbb{R}^n$ to the affine line segment between two distinct points z and z' in \mathbb{R}^n is*

$$\mathcal{D}_{L_2}(s, \text{Line}(z, z')) = \begin{cases} \left\| \vec{z} - \vec{s} + \left(\frac{(\vec{z}' - \vec{z}) \odot (\vec{s} - \vec{z})}{\|\vec{z}' - \vec{z}\|^2} \right) \cdot (\vec{z}' - \vec{z}) \right\| & \text{if } 0 \leq \frac{(\vec{z}' - \vec{z}) \odot (\vec{s} - \vec{z})}{\|\vec{z}' - \vec{z}\|^2} \leq 1 \\ \min(\|\vec{z} - \vec{s}\|, \|\vec{z}' - \vec{s}\|) & \text{otherwise} \end{cases}$$

Moreover, the only point on the line which is $\mathcal{D}_{L_2}(s, \text{Line}(z, z'))$ away from s is $z + \lambda_p \cdot (z' - z)$ with

$$\lambda_p = \frac{(\vec{z}' - \vec{z}) \odot (\vec{s} - \vec{z})}{\|\vec{z}' - \vec{z}\|^2}.$$

Proof. The vector equation of the affine line segment between two points z and z' is $\vec{z} + \lambda \cdot (\vec{z}' - \vec{z})$ with $0 \leq \lambda \leq 1$. By letting λ range over all reals, we get the equation of the infinite line passing through z and z' . Denoting $\vec{z}' - \vec{z} = \vec{v}$, the equation of the line is $\vec{z} + \lambda \cdot \vec{v}$.

Suppose s does not lie on the line $\text{Line}(z, z')$. Let λ_p be such that the vector $\vec{l}_p - \vec{s} = \vec{z} - \vec{s} + \lambda_p \cdot \vec{v}$ is perpendicular to the vector \vec{v} , i.e. $\vec{v} \odot (\vec{l}_p - \vec{s}) = 0$. Intuitively, \vec{v} gives the direction of the line, and the vector from the point s to the point \vec{l}_p is perpendicular to the direction of the line. The distance of the point s from the infinite line is $\|\vec{l}_p - \vec{s}\|$ (see e.g., [Slo01]).

If $0 \leq \lambda_p \leq 1$ then the distance of s to the infinite line segment corresponds to the distance of s to the affine line segment between z and z' . Otherwise, if $\lambda_p < 0$ or $\lambda_p > 1$, we claim the distance

is either $\|\vec{z} - \vec{s}\|$ or $\|\vec{z}' - \vec{s}\|$. This can be seen as follows. It can be easily shown that for any vector \vec{l} on the line that $\|\vec{l} - \vec{s}\|^2 = \|\vec{l}_p - \vec{s}\|^2 + \|\vec{l} - \vec{l}_p\|^2$. Thus,

$$\|\vec{l} - \vec{s}\|^2 = \|\vec{l}_p - \vec{s}\|^2 + (\lambda - \lambda_p)^2 \cdot \|\vec{z}' - \vec{z}\|^2. \quad (15)$$

The minimum is achieved when $|\lambda - \lambda_p|$ is minimum, thus, if $\lambda_p < 0$ or $\lambda_p > 1$, the minimum on the line segment $\text{Line } z, z'$ (*i.e.*, constraining λ to be such that $0 \leq \lambda \leq 1$) is achieved at one of the affine line segment boundaries; if $\lambda_p < 0$ then the point z is closest to s , and if $\lambda_p > 1$ then the point z' is closest to s .

The quantity λ_p can be computed as follows. By definition we have $\vec{v} \odot (\vec{z} - \vec{s} + \lambda_p \cdot \vec{v}) = 0$. Expanding the dot product, $\vec{v} \odot (\vec{z} - \vec{s}) + \lambda_p \cdot \|\vec{v}\|^2 = 0$. Thus, $\lambda_p = \frac{\vec{v} \odot (\vec{s} - \vec{z})}{\|\vec{v}\|^2} = \frac{(\vec{z}' - \vec{z}) \odot (\vec{s} - \vec{z})}{\|\vec{z}' - \vec{z}\|^2}$.

If s lies on the line $\text{Line}(z', z)$, or its infinite extension, then there is no point l_p on the line such that the vector $\vec{l}_p - \vec{s} = \vec{z} - \vec{s} + \lambda_p \cdot (\vec{z}' - \vec{z})$ is perpendicular to the vector $\vec{z}' - \vec{z}$. However, we show the λ_p value computed previously gives the right result. Suppose $s = z + \lambda_s \cdot (z' - z)$. The value of λ_p will be correct if $l_p = s$, *i.e.* if $\lambda_p = \lambda_s$. We have $\lambda_p = \frac{(\vec{z}' - \vec{z}) \odot (\vec{s} - \vec{z})}{\|\vec{z}' - \vec{z}\|^2} = \frac{(\vec{z}' - \vec{z}) \odot (\vec{z} + \lambda_s \cdot (\vec{z}' - \vec{z}) - \vec{z})}{\|\vec{z}' - \vec{z}\|^2}$. Simplifying, we get λ_s .

Putting everything together, we have the desired result. \square

Computation of the least δ such that $\text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z')$ is non-empty. We first show that the computation of the clamping values can be simplified for the L_2 norm. Let $\text{Sphere}(s, \delta)$ denotes the set of points that are exactly δ distance away from s . We show that under certain cases,

$$\min_{\delta \geq 0} \{\delta \mid \text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\} = \inf_{\delta \geq 0} \{\delta \mid \text{Sphere}(s_1, \delta) \cap \text{Sphere}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}.$$

The optimization over the sphere-intersections is an easier problem to solve than the ball-intersections one. The proof strategy for this change of optimization constraints is as follows. Consider the set of points $\text{Ball}(s_1, \delta) \cap \text{Line}(z, z')$; and $\text{Ball}(s_2, \delta) \cap \text{Line}(z, z')$. Each set is a closed set of points on the line $\text{Line}(z, z')$. In terms of the line parameter λ (*i.e.*, where a point on the line is $z + \lambda \cdot (z' - z)$), both sets can be represented as closed subintervals of $[0, 1]$ denoted $[\lambda_1, \lambda'_1]$ and $[\lambda_2, \lambda'_2]$. The least δ in the original optimization problem is the least δ such that these two λ sets intersect. We show that under certain cases, the intersection point corresponds to the boundary of the two balls, and thus, the value is the same as for the sphere-intersection optimization problem (intuitively, the intervals $[\lambda_i, \lambda'_i]$ expand in a strictly monotonic fashion). Moreover, for the cases where this change cannot be made, the ball-intersection optimization problem has a characterization which makes it amenable to solve. We present the formal proof next.

First, we need a technical lemma which characterizes the set $\text{Ball}(s, \delta) \cap \text{Line}(z, z')$ for the norm L_2 . Given two points z, z' , let $\text{Line}_\infty(z, z')$ denote the infinite straight line passing through the two points $z + \lambda \cdot (z' - z)$ for $\lambda \in \mathbb{R}$.

Lemma 5. *Let s and $z \neq z'$ be three points in the space \mathbb{R}^n with the norm L_2 . For $\delta \geq \mathcal{D}(s, \text{Line}(z, z'))$ let $h(\delta) = \{\lambda \mid 0 \leq \lambda \leq 1 \text{ and } \mathcal{D}(s, z + \lambda \cdot (z' - z)) \leq \delta\}$. Let $h_{\max}(\delta) = \sup h(\delta)$, and $h_{\min}(\delta) = \inf h(\delta)$. We have the following facts.*

1. $h(\delta)$ is closed.
2. Over $[\mathcal{D}(s, \text{Line}(z, z')), \mathcal{D}(s, z')]$, we have that (a) $h_{\max}()$ is continuous and strictly increasing, and (b) if $h_{\max}(\delta) = \lambda$, then $\mathcal{D}(s, z + \lambda \cdot (z' - z)) = \delta$.
3. Over $[\mathcal{D}(s, \text{Line}(z, z')), \mathcal{D}(s, z)]$, we have that (a) $h_{\min}(\delta)$ is continuous and strictly decreasing, and (b) if $h_{\min}(\delta) = \lambda$, then $\mathcal{D}(s, z + \lambda \cdot (z' - z)) = \delta$.

Proof. The fact that $h(\delta)$ is closed follows from the fact that $h(\delta)$ is equal to the intersection of $\text{Ball}(s, \delta)$ and $\text{Line}(z, z')$. It can be easily seen that if $h(\delta) \neq \emptyset$, it is of the form $[h_{\min}(\delta), h_{\max}(\delta)]$.

We now prove the second part. Assume $\mathcal{D}(s, z') > \mathcal{D}(s, \text{Line}(z, z'))$ (otherwise the claim is easy to prove). If s does not lie on the line $\text{Line}(z, z')$, we have for any $\lambda \in \mathbb{R}$,

$$\begin{aligned}
\mathcal{D}(s, z + \lambda \cdot (z' - z))^2 &= \|\vec{z} - \vec{s} + \lambda \cdot (\vec{z}' - \vec{z})\|^2 \\
&= \|\vec{z} - \vec{s} + \lambda_p \cdot (\vec{z}' - \vec{z}) + (\lambda - \lambda_p) \cdot (\vec{z}' - \vec{z})\|^2 \text{ where } \lambda_p \in \mathbb{R} \text{ is such that} \\
&\quad \vec{z} - \vec{s} + \lambda_p \cdot (\vec{z}' - \vec{z}) \text{ is perpendicular to } \vec{z}' - \vec{z}. \\
&= \|\vec{z} - \vec{s} + \lambda_p \cdot (\vec{z}' - \vec{z})\|^2 + |\lambda - \lambda_p|^2 \|\vec{z}' - \vec{z}\|^2 \text{ by Equation 15} \\
&= A + |\lambda - \lambda_p|^2 \cdot B \text{ where } A, B \text{ are constants.}
\end{aligned}$$

We comment that λ_p need not necessarily be in $[0, 1]$. If $s = z + \lambda_s(z' - z)$ is on the line $\text{Line}(z, z')$, we have $\mathcal{D}(s, z + \lambda \cdot (z' - z)) = |\lambda_s - \lambda| \cdot \|z' - z\|$, which is equal to $|\lambda_s - \lambda| \cdot \sqrt{B}$. Letting $\lambda_p = \lambda_s$ in this case when $s = z + \lambda_s(z' - z)$, we have that $\mathcal{D}(s, z + \lambda \cdot (z' - z))^2 = A + |\lambda - \lambda_p|^2 \cdot B$ (A is 0 in this case).

Thus, in all cases, for $\delta \geq \mathcal{D}(s, \text{Line}(z, z'))$ (which means $\delta \geq \sqrt{A}$ since $\mathcal{D}(s, \text{Line}(z, z')) \geq \sqrt{A}$), we have, we have

$$h(\delta) = \{\lambda \mid 0 \leq \lambda \leq 1 \text{ such that } |\lambda - \lambda_p| \leq \sqrt{\frac{\delta^2 - A}{B}}\}.$$

Since $\mathcal{D}(s, z') > \mathcal{D}(s, \text{Line}(z, z'))$, we have that $\lambda_p < 1$. This is because we must have by the proof of Lemma 15 and Equation 15, that either $0 \leq \lambda_p < 1$, or $\mathcal{D}(s, \text{Line}(z, z')) = \mathcal{D}(s, z')$ (recall that we assumed $\mathcal{D}(s, \text{Line}(z, z')) < \mathcal{D}(s, z')$). Hence, for $\delta \geq \mathcal{D}(s, \text{Line}(z, z'))$ we have,

$$h_{\max}(\delta) = \min \left(1, \lambda_p + \sqrt{\frac{\delta^2 - A}{B}} \right)$$

Suppose $\mathcal{D}(s, z') \geq \delta \geq \mathcal{D}(s, \text{Line}(z, z'))$. We show $\lambda_p + \sqrt{\frac{\delta^2 - A}{B}} \leq 1$ by contradiction. If $\lambda_p + \sqrt{\frac{\delta^2 - A}{B}} > 1$, then $\delta^2 > A + (1 - \lambda_p)^2 \cdot B = \mathcal{D}(s, z')^2$, i.e. $\mathcal{D}(s, z') < \delta$ which is a contradiction. Thus, for $\mathcal{D}(s, z') \geq \delta \geq \mathcal{D}(s, \text{Line}(z, z'))$, we have

$$h_{\max}(\delta) = \lambda_p + \sqrt{\frac{\delta^2 - A}{B}}. \tag{16}$$

Hence $h_{\max}(\delta)$ is continuous and stictly increasing over $[\mathcal{D}(s, \text{Line}(z, z')), \mathcal{D}(s, z')]$.

If $\delta = \mathcal{D}(s, \text{Line}(z, z'))$, then $h(\delta)$ is just a single point (this can be inferred from the proof of Proposition 15. Thus, for this δ , we have $\mathcal{D}(s, z + h_{\max}(\delta) \cdot (z' - z)) = \delta$. If $\mathcal{D}(s, z') \geq \delta > \mathcal{D}(s, \text{Line}(z, z'))$, then since $h_{\max}()$ is stictly increasing, for all $\mathcal{D}(s, z') \geq \delta > \delta' \geq \mathcal{D}(s, \text{Line}(z, z'))$, we have $h_{\max}(\delta') < h_{\max}(\delta)$, which means that we must have $\mathcal{D}(s, z + h_{\max}(\delta) \cdot (z' - z)) = \delta$. This concludes the proof for h_{\max} . The proof for h_{\min} is similar. \square

Using the previous lemma, we now show that the original optimization problem can be simplified using spheres instead of balls in constraints.

Lemma 6. *Let s_1, s_2, z, z' be four points in the space \mathbb{R}^n with the norm z' . Suppose $\min_{\delta \geq 0} \{\delta \mid \text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}$ is not equal to either $\mathcal{D}(s_1, \text{Line}(z, z'))$ or $\mathcal{D}(s_2, \text{Line}(z, z'))$. Then, $\min_{\delta \geq 0} \{\delta \mid \text{Sphere}(s_1, \delta) \cap \text{Sphere}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}$ exists, and*

$$\min_{\delta \geq 0} \{\delta \mid \text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\} = \min_{\delta \geq 0} \{\delta \mid \text{Sphere}(s_1, \delta) \cap \text{Sphere}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}.$$

Moreover, for the minimum δ , there is only one point in the intersection $\text{Sphere}(s_1, \delta) \cap \text{Sphere}(s_2, \delta) \cap \text{Line}(z, z')$.

Proof. If $s_1 = s_2$, we have $\min_{\delta \geq 0} \{\delta \mid \text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\} = \mathcal{D}(s_1, \text{Line}(z, z'))$, and this is equal to the right hand side of the equality in the lemma. If $z = z'$, then $\min_{\delta \geq 0} \{\delta \mid \text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}$ is equal to $\min_{\delta \geq 0} \{\delta \mid z \in \text{Ball}(s_1, \delta) \text{ and } z \in \text{Ball}(s_2, \delta)\}$ which is equal to $\max(\mathcal{D}(s_1, z), \mathcal{D}(s_2, z))$, and this violates the assumptions of the lemma. Thus, assume $s_1 \neq s_2$ and $z \neq z'$.

Let $\delta^* = \min_{\delta \geq 0} \{\delta \mid \text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}$. We have $\delta^* > 0$ (a value of 0 can arise only if $s_1 = s_2$) Let

$$D_{\min} = \max(\mathcal{D}(s_1, \text{Line}(z, z')), \mathcal{D}(s_2, \text{Line}(z, z'))).$$

Observe that $\delta^* \geq D_{\min}$ by definition, and since δ^* is not equal to either $\mathcal{D}(s_1, \text{Line}(z, z'))$ or $\mathcal{D}(s_2, \text{Line}(z, z'))$, we must have $\delta^* > D_{\min}$.

Consider a δ value $\delta \geq D_{\min}$. Consider the sets $\text{Ball}(s_1, \delta) \cap \text{Line}(z, z')$ and $\text{Ball}(s_2, \delta) \cap \text{Line}(z, z')$. These sets are the $h(\delta)$ sets of Lemma 5. We denote them as $h^{s_1}(\delta)$ and $h^{s_2}(\delta)$ respectively. Using the results from Lemma 5, we have that $h^{s_1}(\delta)$ is the closed interval $[h_{\min}^{s_1}(\delta), h_{\max}^{s_1}(\delta)]$, and similarly for $h^{s_2}(\delta)$. Observe that

$$\text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset \quad \text{iff} \quad [h_{\min}^{s_1}(\delta), h_{\max}^{s_1}(\delta)] \cap [h_{\min}^{s_2}(\delta), h_{\max}^{s_2}(\delta)] \neq \emptyset \quad (17)$$

Also note that by definition, for all $\delta > D_{\min}$, we have $[h_{\min}^{s_1}(\delta), h_{\max}^{s_1}(\delta)] \neq \emptyset$; and also $[h_{\min}^{s_2}(\delta), h_{\max}^{s_2}(\delta)] \neq \emptyset$.

Observe that (a) $[h_{\min}^{s_1}(D_{\min}), h_{\max}^{s_1}(D_{\min})] \neq \emptyset$; and (b) $[h_{\min}^{s_2}(D_{\min}), h_{\max}^{s_2}(D_{\min})] \neq \emptyset$; and (c) $[h_{\min}^{s_1}(D_{\min}), h_{\max}^{s_1}(D_{\min})] \cap [h_{\min}^{s_2}(D_{\min}), h_{\max}^{s_2}(D_{\min})] = \emptyset$, since $\delta^* > D_{\min}$. Consider the case when $h_{\max}^{s_1}(D_{\min}) < h_{\min}^{s_2}(D_{\min})$ (the other case is symmetric), i.e., the interval $[h_{\min}^{s_1}(D_{\min}), h_{\max}^{s_1}(D_{\min})]$ lies to the left of the interval $[h_{\min}^{s_2}(D_{\min}), h_{\max}^{s_2}(D_{\min})]$. Let

$$D_{\max} = \min(\mathcal{D}(s_1, z'), \mathcal{D}(s_2, z)).$$

We claim $\delta^* \leq D_{\max}$. The proof is as follows. By assumption $h_{\max}^{s_1}(D_{\min}) < h_{\min}^{s_2}(D_{\min})$. This means that $h_{\max}^{s_1}(D_{\min}) < 1$, and $h_{\min}^{s_2}(D_{\min}) > 0$. We have the following two cases.

1. $\mathcal{D}(s_1, z') \leq \mathcal{D}(s_2, z)$. Thus $D_{\max} = \mathcal{D}(s_1, z')$ and hence $1 \in h^{s_1}(D_{\max})$, and so $h_{\max}^{s_1}(D_{\max}) = 1$. Using Lemma 5, we have that $D_{\max} > D_{\min}$ since $h_{\max}^{s_1}(D_{\min}) < 1$.
2. $\mathcal{D}(s_2, z) \leq \mathcal{D}(s_1, z')$. Thus $D_{\max} = \mathcal{D}(s_2, z)$ and hence we have $0 \in h^{s_2}(D_{\max})$, and so $0 = h_{\min}^{s_2}(D_{\max})$. Using Lemma 5, we have that $D_{\max} > D_{\min}$ since $h_{\min}^{s_2}(D_{\min}) > 0$.

Thus, in both cases, both the intervals $[h_{\min}^{s_1}(D_{\max}), h_{\max}^{s_1}(D_{\max})]$ and $[h_{\min}^{s_2}(D_{\max}), h_{\max}^{s_2}(D_{\max})]$ are nonempty; and moreover either $h_{\max}^{s_1}(D_{\max}) = 1$ or $0 = h_{\min}^{s_2}(D_{\max})$. This means that the intersection of the intervals $[h_{\min}^{s_1}(D_{\max}), h_{\max}^{s_1}(D_{\max})]$ and $[h_{\min}^{s_2}(D_{\max}), h_{\max}^{s_2}(D_{\max})]$ is non-empty. Using Equation 17, we get that $\delta^* \leq D_{\max}$. Thus,

$$D_{\min} < \delta^* \leq D_{\max}.$$

Consider the functions $h_{\max}^{s_1}()$ and $h_{\min}^{s_2}()$ over the interval $[D_{\min}, D_{\max}]$. The functions are defined on the interval by the conditions of Lemma 5. $h_{\max}^{s_1}()$ is continuous and strictly increasing, and $h_{\min}^{s_2}()$ is continuous and strictly decreasing. At D_{\min} we have $h_{\max}^{s_1}(D_{\min}) < h_{\min}^{s_2}(D_{\min})$, and at D_{\max} we have $h_{\max}^{s_1}(D_{\max}) > h_{\min}^{s_2}(D_{\max})$. Thus these two functions must have the same value at some point in $(D_{\min}, D_{\max}]$, and this point is where the curves of the functions intersect. This intersection point is when $\delta = \delta^*$. The graphs of h_1 and h_2 are depicted in Figure 9. Thus, we have

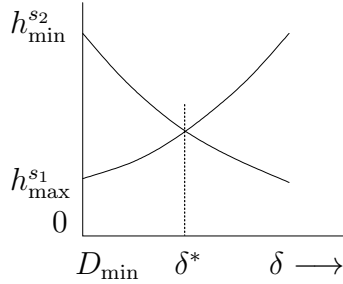


Figure 9: Graphs of $h_{\max}^{s_1}$ and $h_{\min}^{s_2}$.

$h_{\max}^{s_1}(\delta^*) = h_{\min}^{s_2}(\delta^*) = \lambda^*$ (using Lemma 5). Which means that for the point $l^* = z + \lambda^* \cdot (z' - z)$, we have $\mathcal{D}(s_1, l^*) = \delta^*$, and $\mathcal{D}(s_2, l^*) = \delta^*$. Thus

$$\inf_{\delta \geq 0} \{\delta \mid \text{Sphere}(s_1, \delta) \cap \text{Sphere}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\} \geq \delta^*$$

Since we also have

$$\min_{\delta \geq 0} \{\delta \mid \text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\} \leq \inf_{\delta \geq 0} \{\delta \mid \text{Sphere}(s_1, \delta) \cap \text{Sphere}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}$$

and $\min_{\delta \geq 0} \{\delta \mid \text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\} = \delta^*$, we have that $\min_{\delta \geq 0} \{\delta \mid \text{Sphere}(s_1, \delta) \cap \text{Sphere}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}$ exists, and is equal to δ^* , which proves the first part of the lemma. For the second part, we note that the point corresponding to λ^* is the only point on the line $\text{Line}(z, z')$ which is exactly δ^* away from s_1 (or from s_2). This is because the distance of a point on the line given by λ to a point s (not on the line) is $\sqrt{A + |\lambda - \lambda_p|^2 \cdot B}$ where A, B and λ_p are constants (see the proof of Lemma 5); and thus different points on the line have different distances from s . \square

The following corollary states that in a horizontally clamped situation for cells (i, j) and (k, j) , we have only one unique horizontal monotone non-decreasing curve possible which can go from cell (i, j) to (k, j) (see Figure 8 for a horizontally clamped situation). A similar result holds for vertically clamped situations.

Corollary 1. *Given two polygonal curves f, g , let $\delta^1 \geq 0$ be such that in the free space $\text{Free}_{\delta^1}(f, g)$, we have $\text{second}(\mathbf{a}_{i,j}^1) > \text{second}(\mathbf{b}_{k,j}^3)$ for $k > i$; and $\delta^2 \geq 0$ be such that in the free space $\text{Free}_{\delta^2}(f, g)$, we have $\text{second}(\mathbf{a}_{i,j}^1) \leq \text{second}(\mathbf{b}_{k,j}^3)$. Then there exists $\delta^1 < \delta^* \leq \delta^2$, such that in the free space $\text{Free}_{\delta^*}(f, g)$, we have $\text{second}(\mathbf{a}_{i,j}^1) = \text{second}(\mathbf{b}_{k,j}^3)$.*

Proof. For any $\delta \geq 0$, we have $[\text{second}(\mathbf{a}_{i,j}^1), \text{second}(\mathbf{b}_{i,j}^1)]$ in free space $\text{Free}_{\delta}(f, g)$ to be equal to $\text{Ball}(f[i], \delta) \cap g_{[j]}$. Using the proof of Lemma 6, we have that there is a δ^* for which $\text{second}(\mathbf{a}_{i,j}^1) = \text{second}(\mathbf{b}_{k,j}^3)$. \square

Using Lemma 6, we now compute $\delta^* = \min_{\delta \geq 0} \{\delta \mid \text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}$. We restrict our attention to the cases when δ^* is not equal to either $\mathcal{D}(s_1, \text{Line}(z, z'))$ or $\mathcal{D}(s_2, \text{Line}(z, z'))$. It can be checked from the proof of Lemma 6 that this means that s_1, s_2, z, z' are all distinct points. Also, we have that $\delta^* = \min_{\delta \geq 0} \{\delta \mid \text{Sphere}(s_1, \delta) \cap \text{Sphere}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}$.

Given any $\delta > 0$. If a point $z + \lambda \cdot (z' - z)$ on the line $\text{Line}(z, z')$ is in $\text{Sphere}(s_1, \delta) \cap \text{Sphere}(s_2, \delta) \cap \text{Line}(z, z')$, then we must have that it is exactly δ distance away from both s_1 and s_2 . Thus, for some $\delta \in [0, 1]$,

$$\left\| \vec{z} - \vec{s}_1 + \lambda \cdot (\vec{z}' - \vec{z}) \right\| = \left\| \vec{z} - \vec{s}_2 + \lambda \cdot (\vec{z}' - \vec{z}) \right\| \quad (18)$$

$$\text{Thus, } \sum_{k=1}^n (z_k - s_{1,k} + \lambda \cdot (z_k - z'_k))^2 = \sum_{k=1}^n (z_k - s_{2,k} + \lambda \cdot (z_k - z'_k))^2$$

$$\text{Expanding, } \sum_{k=1}^n ((z_k - s_{1,k})^2 + \lambda^2 \cdot (z_k - z'_k)^2 + \lambda \cdot 2 \cdot (z_k - s_{1,k}) \cdot (z_k - z'_k)) =$$

$$\sum_{k=1}^n ((z_k - s_{2,k})^2 + \lambda^2 \cdot (z_k - z'_k)^2 + \lambda \cdot 2 \cdot (z_k - s_{2,k}) \cdot (z_k - z'_k))$$

Cancelling out the common term and rearranging,

$$\lambda \cdot 2 \cdot \left(\sum_{k=1}^n (z_k - s_{1,k}) \cdot (z_k - z'_k) - \sum_{k=1}^n (z_k - s_{2,k}) \cdot (z_k - z'_k) \right) = \sum_{k=1}^n (z_k - s_{2,k})^2 - \sum_{k=1}^n (z_k - s_{1,k})^2$$

Grouping k -terms together,

$$\lambda \cdot 2 \cdot \sum_{k=1}^n ((z_k - s_{1,k}) \cdot (z_k - z'_k) - (z_k - s_{2,k}) \cdot (z_k - z'_k)) = \sum_{k=1}^n ((z_k - s_{2,k})^2 - (z_k - s_{1,k})^2)$$

$$\text{Simplifying, } \lambda \cdot 2 \cdot \sum_{k=1}^n (s_{2,k} - s_{1,k}) \cdot (z_k - z'_k) = \sum_{k=1}^n (s_{2,k}^2 - s_{1,k}^2 + 2 \cdot s_{1,k} \cdot z_k - 2 \cdot s_{2,k} \cdot z_k)$$

$$\text{Simplifying again, } \lambda \cdot 2 \cdot \sum_{k=1}^n (s_{2,k} - s_{1,k}) \cdot (z_k - z'_k) = \sum_{k=1}^n (s_{2,k}^2 - s_{1,k}^2 - 2 \cdot z_k \cdot (s_{2,k} - s_{1,k}))$$

$$\text{A final simplication gives us: } \lambda \cdot 2 \cdot \sum_{k=1}^n (s_{2,k} - s_{1,k}) \cdot (z_k - z'_k) = \sum_{k=1}^n (s_{2,k} - s_{1,k}) \cdot (s_{2,k} + s_{1,k} - 2 \cdot z_k)$$

If $\sum_{k=1}^n (s_{2,k} - s_{1,k}) \cdot (z_k - z'_k)$ is equal to 0, then the value of the LHS in the last equation remains the same for all values of λ . Thus, all vaues of λ satisfy Equation 18. In particular

$\min(\mathcal{D}(s_1, \text{Line}(z, z')), \mathcal{D}(s_2, \text{Line}(z, z')))$. would satisfy Equation 18, and thus δ^* would be either either $\mathcal{D}(s_1, \text{Line}(z, z'))$ or $\mathcal{D}(s_2, \text{Line}(z, z'))$ (recall that $\min(\mathcal{D}(s_1, \text{Line}(z, z')), \mathcal{D}(s_2, \text{Line}(z, z')))$ is a lower bound on δ^*); violating our initial assumption. Thus, $\sum_{k=1}^n (s_{2,k} - s_{1,k}) \cdot (z_k - z'_k)$ is not equal to 0 given the assumptions, and then,

$$\lambda = \frac{\sum_{k=1}^n (s_{2,k} - s_{1,k}) \cdot (s_{2,k} + s_{1,k} - 2 \cdot z_k)}{2 \cdot \sum_{k=1}^n (s_{2,k} - s_{1,k}) \cdot (z_k - z'_k)} \quad (19)$$

Since this is the only value of λ which satisfies Equation 18, we must have $\min_{\delta \geq 0} \{\delta \mid \text{Sphere}(s_1, \delta) \cap \text{Sphere}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}$ to be $\|\vec{z} - \vec{s}_2 + \lambda \cdot (\vec{z}' - \vec{z})\|$ for λ given by Equation 19.

The following function Φ_{L_2} combines everything together giving us Proposition 16

Proposition 16 (Computation of least δ such that $\text{Ball}(q, \delta) \cap \text{Ball}(q', \delta) \cap \text{Line}(z, z')$ is non-empty: L_2). *Given points s_1, s_2, z, z' in \mathbb{R}^n , Function Φ_{L_2} computes $\min_{\delta \geq 0} \{\delta \mid \text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}$ for the L_2 norm.* \square

5.3 L_∞ -norm

We note that the L_∞^S norm is equivalent to the L_∞ norm: $\|\langle s_1, \dots, s_n \rangle, t \rangle\|_{L_\infty^S} = \|\langle s_1, \dots, s_n, t \rangle\|_{L_\infty}$ for $s_i, t \in \mathbb{R}$. Thus, the results in this section are also applicable to the L_∞^S norm.

Proposition 17 (Distance of point to line: L_∞). *The distance from a point $s \in \mathbb{R}^n$ to the affine line segment between two distinct points z and z' in \mathbb{R}^n denoted $\mathcal{D}_{L_\infty}(s, \text{Line}(z, z'))$ is a solution of the following linear program.*

$$\begin{aligned} & \text{minimize } \delta \\ & \text{subject to } z_i - s_i + \lambda \cdot (z'_i - z_i) \leq \delta \quad \text{for } 1 \leq i \leq n \\ & \quad - (z_i - s_i + \lambda \cdot (z'_i - z_i)) \leq \delta \quad \text{for } 1 \leq i \leq n \\ & \quad \delta \geq 0 \\ & \quad 0 \leq \lambda \leq 1 \end{aligned}$$

Proof. The equation of the affine line segment between two points z and z' is $z + \lambda \cdot (z' - z)$ with $0 \leq t \leq 1$. By definition, we have

$$\mathcal{D}_{L_\infty}(s, \text{Line}(z, z')) = \inf_{0 \leq t \leq 1} \max\{|z_i - s_i + \lambda \cdot (z'_i - z_i)| \mid 1 \leq i \leq n\}$$

This can be written as the system of linear constraints in the statement of the lemma, using the fact that $|A| \leq B$ iff $A \leq B$ and $-A \leq B$ for real numbers A, B ; and that $\max\{A_1, \dots, A_n\}$ is by definition the least δ such that $A_i \leq \delta$ for every i . \square

Proposition 18 (Computation of least δ such that $\text{Ball}(q, \delta) \cap \text{Ball}(q', \delta) \cap \text{Line}(z, z')$ is non-empty: L_∞). *Let q, q', z, z' be points in \mathbb{R}^n . The value of $\inf_{\text{Ball}(q, \delta) \cap \text{Ball}(q', \delta) \cap \text{Line}(z, z') \neq \emptyset} \delta$ for the L_∞ norm is*

$$\begin{cases} \mathcal{D}_{L_\infty}(q, \text{Line}(z, z')) & \text{if } q = q' \\ \min(\|q - z\|, \|q' - z\|) & \text{if } q \neq q' \text{ and } z = z' \\ \text{The value of the following linear program 20} & \text{otherwise.} \end{cases}$$

```

Input : Points  $s_1, s_2, z, z'$  in  $\mathbb{R}^n$ 
Output:  $\delta^*$  where  $\delta^* = \min_{\delta \geq 0} \{\delta \mid \text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}$ 
switch  $s_1, s_2, z, z'$  do
  case  $z = z'$  return  $\max(\mathcal{D}(s_1, z), \mathcal{D}(s_2, z))$ ;
  case  $s_1 = s_2$  and  $z \neq z'$  return  $\mathcal{D}(s_1, \text{Line}(z, z'))$ ;
  case  $s_1 \neq s_2$  and  $z \neq z'$ 
    DSet :=  $\emptyset$ ;
    begin
      // Check if  $\delta = \mathcal{D}(s_1, \text{Line}(z, z'))$  is such that
      //  $\text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset$ 
       $D_1 := \mathcal{D}(s_1, \text{Line}(z, z'))$ ;
       $\lambda_1 := \frac{(\vec{z}' - \vec{z}) \odot (\vec{s}_1 - \vec{z})}{\|\vec{z} - \vec{z}'\|_{L_2}^2}$ ; // This value of  $\lambda_1$  is such that the point
      //  $z + \lambda_1 \cdot (z' - z)$  is  $\mathcal{D}(s_1, \text{Line}_\infty(z, z'))$ 
      // distance away from  $s_1$ 
      if  $\lambda_1 < 0$  then  $\lambda_1 := 0$ ; //  $z$  is the closest point on  $\text{Line}(z, z')$  to  $s_1$ 
      else if  $\lambda_1 > 1$  then  $\lambda_1 := 1$ ; //  $z'$  is the closest point on  $\text{Line}(z, z')$  to  $s_1$ 
      if  $\mathcal{D}(s_2, z + \lambda_1 \cdot (z' - z)) \leq D_1$  then DSet :=  $\{D_1\}$ ;
    end
    begin
      // Check if  $\delta = \mathcal{D}(s_2, \text{Line}(z, z'))$  is such that
      //  $\text{Ball}(s_1, \delta) \cap \text{Ball}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset$ 
       $D_2 := \mathcal{D}(s_2, \text{Line}(z, z'))$ ;
       $\lambda_2 := \frac{(\vec{z}' - \vec{z}) \odot (\vec{s}_2 - \vec{z})}{\|\vec{z} - \vec{z}'\|_{L_2}^2}$ ; // This value of  $\lambda_2$  is such that the point
      //  $z + \lambda_2 \cdot (z' - z)$  is  $\mathcal{D}(s_2, \text{Line}_\infty(z, z'))$ 
      // distance away from  $s_2$ 
      if  $\lambda_2 < 0$  then  $\lambda_2 := 0$ ; //  $z$  is the closest point on  $\text{Line}(z, z')$  to  $s_2$ 
      else if  $\lambda_2 > 1$  then  $\lambda_2 := 1$ ; //  $z'$  is the closest point on  $\text{Line}(z, z')$  to  $s_2$ 
      if  $\mathcal{D}(s_1, z + \lambda_2 \cdot (z' - z)) \leq D_2$  then DSet := DSet  $\cup \{D_2\}$ ;
    end
    if DSet  $\neq \emptyset$  then return  $\min(\text{DSet})$ ;
    else return
      
$$\left\| \vec{z} - \vec{s}_1 + \left( \frac{\sum_{k=1}^n (s_{2,k} - s_{1,k}) \cdot (s_{2,k} + s_{1,k} - 2 \cdot z_k)}{2 \cdot \sum_{k=1}^n (s_{2,k} - s_{1,k}) \cdot (z_k - z'_k)} \right) \cdot (\vec{z}' - \vec{z}) \right\|_{L_2}$$

    endsw

```

Function $\Phi_{L_2}(s_1, s_2, z, z')$

$$\begin{aligned}
& \text{minimize } \delta \\
& \text{subject to } z_i - q_i + \lambda \cdot (z'_i - z_i) \leq \delta \quad \text{for } 1 \leq i \leq n \\
& \quad - (z_i - q_i + \lambda \cdot (z'_i - z_i)) \leq \delta \quad \text{for } 1 \leq i \leq n \\
& \quad z_i - q'_i + \lambda \cdot (z'_i - z_i) \leq \delta \quad \text{for } 1 \leq i \leq n \\
& \quad - (z_i - q'_i + \lambda \cdot (z'_i - z_i)) \leq \delta \quad \text{for } 1 \leq i \leq n \\
& \quad \delta \geq 0 \\
& \quad 0 \leq \lambda \leq 1
\end{aligned} \tag{20}$$

Proof. We consider the case when $q \neq q'$ and $z \neq z'$ (the other cases are as in Lemma 14). We want to find the least δ such that there exists some point on $\text{Line}(z, z')$ such that that point is at most δ away from both q and q' . This can be written down as the constraint system 20, using similar reasoning as in the proof of Lemma 17. \square

5.4 L_1^s -norm

In this section, we compute the geometric primitives for the L_2^s -norm defined by $\|\langle s, t \rangle\|_{L_2^s} = \max(\|s\|_{L_2}, |t|)$ for $\langle s, t \rangle \in \mathbb{R}^n \times \mathbb{R}$. We note the following identity that we will use:

$$\left(\|\langle s, t \rangle\|_{L_1^s} \leq \delta \right) \quad \text{iff} \quad (\|s\|_{L_1} \leq \delta \text{ and } |t| \leq \delta) \tag{21}$$

Distance from a point to a Line. Let us be given the points $\langle s, t_s \rangle, \langle z, t_z \rangle, \langle z', t_{z'} \rangle$ with $s, z, z' \in \mathbb{R}^n$ and $t \in \mathbb{R}$. For the L_1^s -norm, the distance $\mathcal{D}_{L_1^s}(\langle s, t_s \rangle, \text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle))$ is equal to the following

$$\begin{aligned}
\mathcal{D}_{L_1^s}(\langle s, t_s \rangle, \text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle)) &= \inf_{0 \leq \lambda \leq 1} \|\langle z, t_z \rangle - \langle s, t_s \rangle + \lambda \cdot (\langle z', t_{z'} \rangle - \langle z, t_z \rangle)\|_{L_1^s} \\
&= \inf_{0 \leq \lambda \leq 1} \left(\max \left(\|z - s + \lambda \cdot (z' - z)\|_{L_1}, |t_z - t_s + \lambda \cdot (t_{z'} - t_z)| \right) \right) \\
&= \inf_{0 \leq \lambda \leq 1} \left\{ Z \mid \begin{array}{l} Z \geq \|z - s + \lambda \cdot (z' - z)\|_{L_1} \text{ and } \\ Z \geq |t_z - t_s + \lambda \cdot (t_{z'} - t_z)| \end{array} \right\}
\end{aligned}$$

This can be written as the following optimization problem.

$$\begin{aligned}
& \text{minimize } Z \\
& \text{subject to } \left(\sum_{i=1}^n |z_i - s_i + \lambda \cdot (z'_i - z_i)| \right) - Z \leq 0 \\
& \quad (|t_z - t_s + \lambda \cdot (t_{z'} - t_z)|) - Z \leq 0 \\
& \quad 0 \leq \lambda \leq 1 \\
& \quad Z \geq 0
\end{aligned}$$

The above optimization problem can then be written as:

$$\begin{aligned}
& \text{minimize} && Z \\
& \text{subject to} && \left(\sum_{i=1}^n |U_i| \right) - Z \leq 0 \\
& && (|U_t|) - Z \leq 0 \\
& && U_i = z_i - s_i + \lambda \cdot (z' - z) \quad \text{for } 1 \leq i \leq n \\
& && U_t = t_z - t_s + \lambda \cdot (t_{z'} - t_z) \\
& && 0 \leq \lambda \leq 1 \\
& && Z \geq 0
\end{aligned}$$

We remove the absolute values in the constraints as follows. We introduce $n + 1$ new variables $absU_i$ for $1 \leq i \leq n$, and $absU_t$ such that $absU_i \geq U_i$ and $absU_i \geq -U_i$ and similarly for $absU_t$. Consider the linear program

$$\begin{aligned}
& \text{minimize} && Z \\
& \text{subject to} && \left(\sum_{i=1}^n absU_i \right) - Z \leq 0 \\
& && absU_i - Z \leq 0 \\
& && U_i - absU_i \leq 0 \quad \text{for } 1 \leq i \leq n \\
& && -U_i - absU_i \leq 0 \quad \text{for } 1 \leq i \leq n \\
& && U_t - absU_t \leq 0 \\
& && -U_t - absU_t \leq 0 \\
& && absU_i \geq 0 \quad \text{for } 1 \leq i \leq n \\
& && absU_t \geq 0 \\
& && U_i = z_i - s_i + \lambda \cdot (z' - z) \quad \text{for } 1 \leq i \leq n \\
& && U_t = t_z - t_s + \lambda \cdot (t_{z'} - t_z) \\
& && 0 \leq \lambda \leq 1 \\
& && Z \geq 0
\end{aligned} \tag{22}$$

Note that $absU_i \geq U_i$ and $absU_i \geq -U_i$ only ensure $absU_i \geq |U_i|$, equality is *not* guaranteed. However, the new constraint $\sum_{i=1}^n absU_i - Z \leq 0$ ensures that $absU_i$ can be taken to be $|U_i|$ in the computation of the optimum in the linear program. The proof of correctness is as for the proof of correctness of the linear program 14.

The results of the proceeding discussion are summarized in the following proposition.

Proposition 19 (Distance of point to Line: L_1^S). *Let $\langle s, t_s \rangle, \langle z, t_z \rangle, \langle z', t_{z'} \rangle$ be points with $s, z, z' \in \mathbb{R}^n$ and $t \in \mathbb{R}$. For the L_1^S -norm, the distance of the point $\langle s, t_s \rangle$ from the line $\text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle)$ is the value of the linear program 22.* \square

Computation of the least δ such that $\text{Ball}(\langle s_1, t_{s_1} \rangle, \delta) \cap \text{Ball}(\langle s_2, t_{s_1} \rangle, \delta) \cap \text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle)$ is non-empty.

The value of the least δ is specified by the following optimization problem.

$$\begin{aligned}
& \text{minimize } \delta \\
& \text{subject to } \left\| \langle z, t_z \rangle - \langle s_1, t_{s_1} \rangle + \lambda \cdot (\langle z', t_{z'} \rangle - \langle z, t_z \rangle) \right\|_{L_1^s} \leq \delta \\
& \quad \left\| \langle z, t_z \rangle - \langle s_2, t_{s_2} \rangle + \lambda \cdot (\langle z', t_{z'} \rangle - \langle z, t_z \rangle) \right\|_{L_1^s} \leq \delta \\
& \quad 0 \leq \lambda \leq 1 \\
& \quad 0 \leq \delta
\end{aligned}$$

Expanding the L_1^s norm, and using Equation 21,

$$\begin{aligned}
& \text{minimize } \delta \\
& \text{subject to } \left(\sum_{i=1}^n |z_i - s_{1,i} + \lambda \cdot (z'_i - z_i)| \right) - \delta \leq 0 \\
& \quad |t_z - t_{s_1} + \lambda \cdot (t_{z'} - t_z)| - \delta \leq 0 \\
& \quad \left(\sum_{i=1}^n |z_i - s_{2,i} + \lambda \cdot (z'_i - z_i)| \right) - \delta \leq 0 \\
& \quad |t_z - t_{s_2} + \lambda \cdot (t_{z'} - t_z)| - \delta \leq 0 \\
& \quad 0 \leq \lambda \leq 1 \\
& \quad 0 \leq \delta
\end{aligned}$$

As was done in the case of the distance of a point from a line, we rewrite the above constraint system as:

$$\begin{aligned}
& \text{minimize } \delta \\
& \text{subject to } \left(\sum_{i=1}^n |U_i| \right) - \delta \leq 0 \\
& \quad |U_t| - \delta \leq 0 \\
& \quad \left(\sum_{i=1}^n |U'_i| \right) - \delta \leq 0 \\
& \quad |U'_t| - \delta \leq 0 \\
& \quad U_i = z_i - s_{1,i} + \lambda \cdot (z'_i - z_i) \quad \text{for } 1 \leq i \leq n \\
& \quad U_t = t_z - t_{s_1} + \lambda \cdot (t_{z'} - t_z) \\
& \quad U'_i = z_i - s_{1,i} + \lambda \cdot (z'_i - z_i) \quad \text{for } 1 \leq i \leq n \\
& \quad U'_t = t_z - t_{s_2} + \lambda \cdot (t_{z'} - t_z) \\
& \quad 0 \leq \lambda \leq 1 \\
& \quad 0 \leq \delta
\end{aligned}$$

We remove the absolute values using the same trick as before, by introducing $2 \cdot n + 2$ new variables $absU_i, absU'_i$ for $1 \leq i \leq n$, and $absU_t$ and $absU'_t$. The proof of correctness of this

transformation is exactly the same as it was for the case of the distance of a point from a line.

$$\begin{aligned}
& \text{minimize } \delta \\
& \text{subject to } \left(\sum_{i=1}^n \text{abs}U_i \right) - \delta \leq 0 \\
& \quad \text{abs}U_t - \delta \leq 0 \\
& \quad \left(\sum_{i=1}^n \text{abs}U'_i \right) - \delta \leq 0 \\
& \quad \text{abs}U'_t - \delta \leq 0 \\
& \quad U_i = z_i - s_{1,i} + \lambda \cdot (z'_i - z_i) \quad \text{for } 1 \leq i \leq n \\
& \quad U_t = t_z - t_{s_1} + \lambda \cdot (t_{z'} - t_z) \\
& \quad U'_i = z_i - s_{1,i} + \lambda \cdot (z'_i - z_i) \quad \text{for } 1 \leq i \leq n \\
& \quad U'_t = t_z - t_{s_2} + \lambda \cdot (t_{z'} - t_z) \\
& \quad U_i - \text{abs}U_i \leq 0 \quad \text{for } 1 \leq i \leq n \\
& \quad -U_i - \text{abs}U_i \leq 0 \quad \text{for } 1 \leq i \leq n \\
& \quad U_t - \text{abs}U_t \leq 0 \\
& \quad -U_t - \text{abs}U_t \leq 0 \\
& \quad U'_i - \text{abs}U'_i \leq 0 \quad \text{for } 1 \leq i \leq n \\
& \quad -U'_i - \text{abs}U'_i \leq 0 \quad \text{for } 1 \leq i \leq n \\
& \quad U'_t - \text{abs}U'_t \leq 0 \\
& \quad -U'_t - \text{abs}U'_t \leq 0 \\
& \quad \text{abs}U_i \geq 0 \quad \text{for } 1 \leq i \leq n \\
& \quad \text{abs}U_t \geq 0 \\
& \quad \text{abs}U'_i \geq 0 \quad \text{for } 1 \leq i \leq n \\
& \quad \text{abs}U'_t \geq 0 \\
& \quad 0 \leq \lambda \leq 1 \\
& \quad 0 \leq \delta
\end{aligned} \tag{23}$$

Proposition 20. *Let $\langle s_1, t_{s_1} \rangle, \langle s_2, t_{s_2} \rangle, \langle z, t_z \rangle, \langle z', t_{z'} \rangle$ be points with $s_1, s_2, z, z' \in \mathbb{R}^n$ and $t_{s_1}, t_{s_2}, t_z, t_{z'} \in \mathbb{R}$. The value of the least $\delta \geq 0$ such that $\text{Ball}(\langle s_1, t_{s_1} \rangle, \delta) \cap \text{Ball}(\langle s_2, t_{s_2} \rangle, \delta) \cap \text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle)$ is non-empty is given by the linear program 23. \square*

5.5 L_2^s -norm

In this section, we compute the geometric primitives for the L_2^s -norm defined by $\|\langle s, t \rangle\|_{L_2^s} = \max(\|s\|_{L_2}, |t|)$ for $\langle s, t \rangle \in \mathbb{R}^n \times \mathbb{R}$. We note the following identity that we will use:

$$(\|\langle s, t \rangle\|_{L_2^s} \leq \delta) \quad \text{iff} \quad (\|s\|_{L_2} \leq \delta \text{ and } |t| \leq \delta) \tag{24}$$

Distance from a point to a Line. Let us be given the points $\langle s, t_s \rangle, \langle z, t_z \rangle, \langle z', t_{z'} \rangle$ with $s, z, z' \in \mathbb{R}^n$ and $t \in \mathbb{R}$. For the L_2^s -norm, the distance $\mathcal{D}_{L_2^s}(\langle s, t_s \rangle, \text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle))$ is equal to the

following using Equation 24:

$$\min_{\delta \geq 0} \left(\text{There exists } \lambda \in [0, 1] \text{ such that } \left\{ \begin{array}{c} |t_z - t_s + \lambda \cdot (t_{z'} - t_z)| \leq \delta \\ \text{and} \\ \left\| \vec{z} - \vec{s} + \lambda \cdot (\vec{z}' - \vec{z}) \right\|_{L_2} \leq \delta \end{array} \right\} \right) \quad (25)$$

Note that the minimum exists because $\lambda \in [0, 1]$, a closed set. To compute this quantity, we follow a strategy similar to that for computing the optimization problem in L_2 of two balls intersecting with a line – we reason over λ sets. We have the following cases.

- If $\langle z, t_z \rangle = \langle z', t_{z'} \rangle$, the minimal δ is just $\max(\|z - s\|_{L_2}, |t_z - t_s|)$. So, assume $\langle z, t_z \rangle \neq \langle z', t_{z'} \rangle$.
- If $\langle z, t_z \rangle \neq \langle z', t_{z'} \rangle$, but $z = z'$, then we show that Equation 25 has the value $D = \max(\|z - s\|_{L_2}, \mathcal{D}(t_s, \text{Line}(t_z, t_{z'})))$ as follows.

$D \geq \mathcal{D}_{L_2^s}(\langle s, t_s \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle))$: We show there exists a $\lambda \in [0, 1]$ such that both the clauses of Equation 25 are satisfied. By definition, we have that there exists a $\lambda \in [0, 1]$ such that $|t_z - t_s + \lambda \cdot (t_{z'} - t_z)| \leq D$. Moreover, this value of λ gives the point z on the line $\text{Line}(z, z)$ (both endpoints are the same). And by definition, we have $\|z - s\|_{L_2} \leq D$. This proves $D \geq \mathcal{D}_{L_2^s}(\langle s, t_s \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle))$.

$D \leq \mathcal{D}_{L_2^s}(\langle s, t_s \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle))$: Let λ be the value corresponding to the optimal δ according to Equation 25. This means that $\delta \geq \|z - s\|_{L_2}$ (from the second clause, and noting $z + \lambda(z - z) = z$). Also, $\delta \geq \mathcal{D}(t_s, \text{Line}(t_z, t_{z'}))$, otherwise the first clause cannot be satisfied for any $\lambda \in [0, 1]$. This proves $D \leq \mathcal{D}_{L_2^s}(\langle s, t_s \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle))$.

- If $\langle z, t_z \rangle \neq \langle z', t_{z'} \rangle$, and $z \neq z'$, but $t_z = t_{z'}$, we claim that Equation 25 has the value $\max(\mathcal{D}_{L_2}(s, \text{Line}(z, z')), |t_s - t_z|)$. The proof is similar to the previous case.

- Assume $z \neq z'$ and $t_z \neq t_{z'}$. First we compute $\mathcal{D}_{L_2}(s, \text{Line}(z, z'))$ as in Proposition 15. We also compute λ_p , such that the point $z + \lambda_p \cdot (z' - z)$ is $\mathcal{D}_{L_2}(s, \text{Line}(z, z'))$ distance away from s . This can be done as done in the proof of Proposition 15: we compute $\frac{(\vec{z}' - \vec{z}) \odot (\vec{s} - \vec{z})}{\|\vec{z}' - \vec{z}\|^2}$; and (a) if this value is in the interval $[0, 1]$, then it is λ_p ; (b) If this value is less than 0, then $\lambda_p = 0$ (in this case z is the closest point to s); (c) If this value is greater than 1, then $\lambda_p = 1$ (in this case z' is the closest point to s).

Let $t_s = t_z + \lambda_{t_s} \cdot (t_{z'} - t_z)$. Thus, $\lambda_{t_s} = \frac{t_s - t_z}{t_{z'} - t_z}$ (note that by assumption, $t_z \neq t_{z'}$). Note that λ_{t_s} will in general *not* be in the interval $[0, 1]$. Assume $\lambda_{t_s} \geq \lambda_p$ (otherwise, we just switch $\langle z, t_z \rangle, \langle z, t_{z'} \rangle$). Thus on the real line, the point where λ_{t_s} lies is to the right of the minimum distance point λ_p which minimizes the L_2 norm for the \mathbb{R}^n components. The value λ_{t_s} can thus be assumed to be in the interval $[0, \infty)$.

Let $h^1 : [\mathcal{D}_{L_2}(s, \text{Line}(z, z')), \infty) \mapsto [0, 1]$ be the function $h^1(\delta) = \{\lambda \mid 0 \leq \lambda \leq 1 \text{ and } \|z - s + \lambda \cdot (z' - z)\|_{L_2} \leq \delta\}$; thus, $h^1()$ is the $h()$ function from Lemma 5. The value $h^1(\delta)$ indicates the range of λ values in $[0, 1]$ such that for these λ values, the points $z + \lambda \cdot (z' - z)_{L_2}$ are at most δ away (in the L_2 metric) from s . Note that for $\delta \geq \mathcal{D}_{L_2}(s, \text{Line}(z, z'))$, the set $h^1(\delta)$ is never empty. Let $h^t : \mathbb{R}_+ \mapsto \mathbb{R}$ be the function $h^t(\delta) = \{\lambda \mid |t_z - t_s + \lambda \cdot (t_{z'} - t_z)| \leq \delta\}$, i.e. the λ values in \mathbb{R} such that the corresponding point $t_z + \lambda \cdot (t_{z'} - t_z)$ on the infinite line is at most δ away from t_s . Observe that the solution to Equation 25 is $\min_{\delta \geq \mathcal{D}_{L_2}(s, \text{Line}(z, z'))} h^1(\delta) \cap h^t(\delta) \neq \emptyset$. We take $\delta \geq \mathcal{D}_{L_2}(s, \text{Line}(z, z'))$ because clearly for $\delta < \mathcal{D}_{L_2}(s, \text{Line}(z, z'))$ there is no $\lambda \in [0, 1]$ such that $\left\| \vec{z} - \vec{s} + \lambda \cdot (\vec{z}' - \vec{z}) \right\|_{L_2} \leq \delta$. Let $h_{\max}^1(\delta)$ be as in Lemma 5, i.e. $h_{\max}^1(\delta) = \min h^1(\delta)$ (the

maximum exists since $h^1(\delta)$ is closed). Let $h_{\min}^t(\delta) = \min h^t(\delta)$ (the minimum exists since $h^t(\delta)$ is closed). We thus have the solution to Equation 25 to be

$$\min_{\delta \geq \mathcal{D}_{L_2}(s, \text{Line}(z, z'))} \text{ such that } h_{\max}^1(\delta) \geq h_{\min}^t(\delta). \quad (26)$$

We have $h^t(\delta)$, which is defined to be $\{\lambda \mid |t_z - t_s + \lambda \cdot (t_{z'} - t_z)| \leq \delta\}$, to be equal to:

$$\begin{aligned} & \{\lambda \mid |t_z - (t_z + \lambda_s \cdot (t_{z'} - t_z)) + \lambda \cdot (t_{z'} - t_z)| \leq \delta\} \text{ since } t_s = t_z + \lambda_s \cdot (t_{z'} - t_z) \\ &= \{\lambda \mid |(\lambda - \lambda_s) \cdot (t_{z'} - t_z)| \leq \delta\} \\ &= \{\lambda \mid |(\lambda - \lambda_s)| \cdot |t_{z'} - t_z| \leq \delta\} \\ &= \{\lambda \mid |(\lambda - \lambda_s)| \leq \frac{\delta}{|t_{z'} - t_z|}\} \text{ note that by assumption } t_z \neq t_{z'} \end{aligned}$$

Thus, $h_{\min}^t(\delta)$ is $\lambda_s - \frac{\delta}{|t_{z'} - t_z|}$; and since $\lambda_s = \frac{t_s - t_z}{t_{z'} - t_z}$, we have

$$h_{\min}^t(\delta) = \frac{t_s - t_z}{t_{z'} - t_z} - \frac{\delta}{|t_{z'} - t_z|} \quad (27)$$

Observe that $h_{\min}^t(\delta)$ is continuous and strictly decreasing; and that if $h_{\min}^t(\delta) = \lambda_\delta$, then $|t_z - t_s + \lambda_\delta \cdot (t_{z'} - t_z)| = \delta$. We have the following cases.

- (a) $h_{\min}^t(\mathcal{D}_{L_2}(s, z')) > 1$. This means that $\min_{\delta \geq \mathcal{D}_{L_2}(s, \text{Line}(z, z'))} h_{\max}^1(\delta) \geq h_{\min}^t(\delta)$ is equal to $\min_{\delta > \mathcal{D}_{L_2}(s, z')} h_{\max}^1(\delta) \geq h_{\min}^t(\delta)$ (since the maximum value of $h_{\max}^1()$ is 1, and h_{\min}^t is strictly decreasing). Since $h_{\max}^1(\delta) = 1$ for all $\delta \geq \mathcal{D}_{L_2}(s, z')$, we have:

$$\left(\min_{\delta > \mathcal{D}_{L_2}(s, z')} \text{ such that } h_{\max}^1(\delta) \geq h_{\min}^t(\delta) \right) = \left(\min_{\delta \geq 0} \text{ such that } h_{\min}^t(\delta) \leq 1 \right).$$

Using Equation 27, we get the above minimum to be when $\frac{t_s - t_z}{t_{z'} - t_z} - \frac{\delta}{|t_{z'} - t_z|} = 1$, i.e., the minimum value for δ is $\delta = |t_{z'} - t_z| \cdot \left(\frac{t_s - t_z}{t_{z'} - t_z} - 1 \right)$.

- (b) $h_{\min}^t(\mathcal{D}_{L_2}(s, z')) \leq 1$. Note that $h_{\max}^1(\mathcal{D}_{L_2}(s, z')) = 1$, thus, the solution to Equation 26 must be $\delta \leq 1$. Hence, we can restrict the range of Equation 26 as follows.

$$\min_{\mathcal{D}_{L_2}(s, z') \geq \delta \geq \mathcal{D}_{L_2}(s, \text{Line}(z, z'))} \text{ such that } h_{\max}^1(\delta) \geq h_{\min}^t(\delta). \quad (28)$$

Suppose $h_{\min}^t(\mathcal{D}_{L_2}(s, \text{Line}(z, z')))$ has the value λ such that $\lambda \leq \lambda_p$ where λ_p is such that $\mathcal{D}_{L_2}(z - s + \lambda_p \cdot (z' - z))$ is equal to $\mathcal{D}_{L_2}(s, \text{Line}(z, z'))$, that is λ_p gives the point on the line $\text{Line}(z, z')$ which minimizes the distance to s . In this case, the minimum value of δ satisfying Equation 28 is $\mathcal{D}_{L_2}(s, \text{Line}(z, z'))$.

Suppose $h_{\min}^t(\mathcal{D}_{L_2}(s, \text{Line}(z, z')))$ has the value λ such that $\lambda > \lambda_p$ where λ_p is as before. Over the range $\mathcal{D}_{L_2}(s, z') \geq \delta \geq \mathcal{D}_{L_2}(s, \text{Line}(z, z'))$, the function $h_{\max}^1()$ is continuous and strictly increasing by Lemma 5. Also, the function $h_{\min}^t()$ is continuous and strictly decreasing. At $\delta = \mathcal{D}_{L_2}(s, \text{Line}(z, z'))$, the value of $h_{\min}^t(\delta)$ is greater than that of $h_{\max}^1(\delta)$. And at $\delta' = \mathcal{D}_{L_2}(s, z')$, the value of $h_{\min}^t(\delta')$ is greater than that of $h_{\max}^1(\delta')$. Thus, $h_{\min}^t()$ and $h_{\max}^1()$ must intersect at

some point δ^* in between δ and δ' . Let $h_{\min}^t(\delta^*) = \lambda^* = h_{\max}^1(\max)$. Moreover, by the properties of $h_{\min}^t()$ and $h_{\max}^1()$, we have the following two equalities: $\mathcal{D}_{L_2}(z - s + \lambda^* \cdot (z' - z)) = \delta^*$; and $|t_z - t_s + \lambda^* \cdot (t_{z'} - t_z)| = \delta^*$. Thus,

$$\begin{aligned} \|(z - s + \lambda^* \cdot (z' - z))\|^2 &= (t_z - t_s + \lambda^* \cdot (t_{z'} - t_z))^2 \\ \text{Expanding, } \sum_{k=1}^n \left((z_k - s_k)^2 + (\lambda^*)^2 \cdot (z'_k - z_k)^2 + \lambda^* \cdot 2 \cdot (z_k - s_k) \cdot (z'_k - z_k) \right) &= \\ \sum_{k=1}^n \left((t_{z,k} - t_{s,k})^2 + (\lambda^*)^2 \cdot (t_{z',k} - t_{z,k})^2 + \lambda^* \cdot 2 \cdot (t_{z,k} - t_{s,k}) \cdot (z'_k - z_k) \right) & \\ \text{Rearranging, } (\lambda^*)^2 \cdot \sum_{k=1}^n ((z'_k - z_k)^2 - (t_{z',k} - t_{z,k})^2) + & \\ \lambda^* \cdot 2 \cdot \sum_{k=1}^n ((z_k - s_k) \cdot (z'_k - z_k) - (t_{z,k} - t_{s,k}) \cdot (z'_k - z_k)) + & \\ \sum_{k=1}^n ((z_k - s_k)^2 - (t_{z,k} - t_{s,k})^2) &= 0 \end{aligned}$$

We solve the quadratic equation above to obtain λ^* . The assumed conditions imply that there will be exactly one root in the interval $[0, 1]$ (otherwise we would get two intersections of $h_{\min}^t()$ and $h_{\max}^1()$, which is not possible since the first one is strictly decreasing, and the second one strictly increasing. The optimal value δ^* is then $|t_z - t_s + \lambda^* \cdot (t_{z'} - t_z)|$.

We put everything together in Function $\Psi_{L_2^S}$.

Proposition 21 (Distance of point to line: L_2^S). *Given points $\langle s, t_s \rangle, \langle z, t_z \rangle, \langle z', t_{z'} \rangle$ in $\mathbb{R}^n \times \mathbb{R}$, Function $\Psi_{L_2^S}$ computes the minimum distance of the point $\langle s, t_s \rangle$ from the line $\text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle)$ for the L_2^S -norm. \square*

Computation of the least δ such that $\text{Ball}(\langle s_1, t_{s_1} \rangle, \delta) \cap \text{Ball}(\langle s_2, t_{s_2} \rangle, \delta) \cap \text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle)$ is non-empty.

We use the following result to solve the optimization problem.

Proposition 22 (Helley's theorem [Eck93]). *Let X_1, \dots, X_r be a finite collection of convex subsets of \mathbb{R}^d with $r > d$. If the intersection of every $d+1$ of these sets is nonempty, then $\bigcap_{i=1}^r X_i \neq \emptyset$. \square*

Let us be given the points $\langle s_1, t_{s_1} \rangle, \langle s_2, t_{s_2} \rangle, \langle z, t_z \rangle, \langle z', t_{z'} \rangle$ with $s_1, s_2, z, z' \in \mathbb{R}^n$ and $t_{s_1}, t_{s_2}, t_z, t_{z'} \in \mathbb{R}$. For the L_2^S -norm, given a $\delta \geq 0$, we have $\text{Ball}(\langle s_1, t_{s_1} \rangle, \delta) \cap \text{Ball}(\langle s_2, t_{s_2} \rangle, \delta) \cap \text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle)$ to be non-empty iff there is some point $\langle l, t_l \rangle$ on the line $\text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle)$ such that both the following two conditions hold: (i) $\|\langle s_1, t_{s_1} \rangle - \langle l, t_l \rangle\|_{L_2^S} \leq \delta$; and (ii) $\|\langle s_2, t_{s_2} \rangle - \langle l, t_l \rangle\|_{L_2^S} \leq \delta$; and Using 21, the least δ such that the previous two conditions

```

Input : Points  $\langle s, t_s \rangle, \langle z, t_z \rangle, \langle z, t_{z'} \rangle$  in  $\mathbb{R}^n \times \mathbb{R}$ 
Output:  $\mathcal{D}_{L_2^s}(\langle s, t_s \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle))$ 
switch  $\langle z, t_z \rangle, \langle z, t_{z'} \rangle$  do
  case  $\langle z, t_z \rangle = \langle z', t_{z'} \rangle$  return  $\max(\|z - s\|_{L_2}, |t_z - t_s|)$ ;
  case  $\langle z, t_z \rangle = \langle z', t_{z'} \rangle$  and  $z = z'$  return  $\max(\|z - s\|_{L_2}, \mathcal{D}(t_s, \text{Line}(t_z, t_{z'})))$ ;
  case  $\langle z, t_z \rangle \neq \langle z', t_{z'} \rangle$ , and  $z \neq z'$ , and  $t_z = t_{z'}$  return
     $\max(\mathcal{D}_{L_2}(s, \text{Line}(z, z')), |t_s - t_z|)$ ;
  case  $z \neq z'$ , and  $t_z \neq t_{z'}$ 
     $\lambda_p := \frac{(\vec{z}' - \vec{z}) \odot (\vec{s} - \vec{z})}{\|\vec{z}' - \vec{z}\|^2}$ ;
    if  $\lambda_p < 0$  then  $\lambda_p := 0$ ;
    else if  $\lambda_p > 1$  then  $\lambda_p := 1$ ;
     $\lambda_{t_s} := \frac{t_s - t_z}{t_{z'} - t_z}$ ;
    if  $\lambda_{t_s} < \lambda_p$  then
      swap  $\langle z, t_z \rangle$  and  $\langle z, t_{z'} \rangle$ ; // Make  $\lambda_{t_s} \geq \lambda_p$ 
       $\lambda_p := 1 - \lambda_p$ ;
       $\lambda_{t_s} := 1 - \lambda_{t_s}$ ;
    end
     $\alpha := \left( \frac{t_s - t_z}{t_{z'} - t_z} - \frac{\|s - z'\|_{L_2}}{|t_{z'} - t_z|} \right)$ ; //  $\alpha = h_{\min}^t(\mathcal{D}_{L_2}(s, z'))$ 
    if  $\alpha > 1$  then return  $|t_{z'} - t_z| \cdot \left( \frac{t_s - t_z}{t_{z'} - t_z} - 1 \right)$ ;
    else
      if  $\alpha \leq \lambda_p$  then return  $\mathcal{D}_{L_2}(s, \text{Line}(z, z'))$ ;
      else
        Solve the following quadratic equation for  $\lambda^* \in [0, 1]$ 

$$\begin{aligned}
 & (\lambda^*)^2 \cdot \sum_{k=1}^n ((z'_k - z_k)^2 - (t_{z',k} - t_{z,k})^2) + \\
 & \lambda^* \cdot 2 \cdot \sum_{k=1}^n ((z_k - s_k) \cdot (z'_k - z_k) - (t_{z,k} - t_{s,k}) \cdot (z'_k - z_k)) + \\
 & \sum_{k=1}^n ((z_k - s_k)^2 - (t_{z,k} - t_{s,k})^2) = 0
 \end{aligned}$$

        return  $|t_z - t_s + \lambda^* \cdot (t_{z'} - t_z)|$ ;
      end
    end
  endsw
endsw

```

Function $\Psi_{L_2^s}(\langle s, t_s \rangle, \langle z, t_z \rangle, \langle z, t_{z'} \rangle)$

hold is thus given by the following expression:

$$\min_{\delta \geq 0} \left(\text{There exists } \lambda \in [0, 1] \text{ such that } \left\{ \begin{array}{l} \left\| \vec{z} - \vec{s}_1 + \lambda \cdot (\vec{z}' - \vec{z}) \right\|_{L_2} \leq \delta; \text{ and} \\ |t_z - t_{s_1} + \lambda \cdot (t_{z'} - t_z)| \leq \delta; \text{ and} \\ \left\| \vec{z} - \vec{s}_2 + \lambda \cdot (\vec{z}' - \vec{z}) \right\|_{L_2} \leq \delta; \text{ and} \\ |t_z - t_{s_2} + \lambda \cdot (t_{z'} - t_z)| \leq \delta \end{array} \right\} \right) \quad (29)$$

A possible approach to solve the above optimization problem is to proceed as in L_2 , and reason over λ sets. However, in this case we have 4 λ sets at play, so at first glance we cannot just use the old approach. However, we show that Helly's theorem 22 allows us to restrict our attention to only two λ sets at a time. We proceed as follows.

Symmetry: Observe from Equation 29 that we can swap t_{s_1} and t_{s_2} without changing the value of the least δ , *i.e.*, the minimum δ such that $\text{Ball}(\langle s_1, t_{s_1} \rangle, \delta) \cap \text{Ball}(\langle s_2, t_{s_2} \rangle, \delta) \cap \text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle)$ is non-empty equals the minimum δ such that $\text{Ball}(\langle s_1, t_{s_2} \rangle, \delta) \cap \text{Ball}(\langle s_2, t_{s_1} \rangle, \delta) \cap \text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle)$ is non-empty. We will use this symmetry to reduce the number of cases that need to be considered.

Swapping $\langle z, t_z \rangle$ with $\langle z', t_{z'} \rangle$: Clearly we can swap $\langle z, t_z \rangle$ with $\langle z', t_{z'} \rangle$, as the line between the two points remains the same. Observe that if $t = t_z + \lambda \cdot (t_{z'} - t_z)$, then we can rewrite the expression as $t = t_{z'} + (1 - \lambda) \cdot (t_z - t_{z'})$ (and similarly for z, z'), thus after the swap, the new λ values are 1 minus the old ones.

The function h^∞ : Recall the function $h(\delta)$ from Lemma 5. We define a similar function which can have all elements from reals, not just from $[0, 1]$. Suppose we are given a infinite line $\text{Line}_\infty(l_1, l_2)$, passing through l_1 and l_2 ; and a point l (not necessarily on that line). Given $\delta \geq 0$, we want a representation of all points on the line that are at most δ away from l .

$$h_l^{\infty, l_1, l_2}(\delta) = \{\lambda \mid \|l_1 - l + \lambda \cdot (l_2 - l_1)\| \leq \delta\}$$

We omit the annotations l_1, l_2 which specify the line when the line is clear from context. For the norms L_2 and $|\cdot|$, it can be show that for $\delta \geq \mathcal{D}(\text{Line}(l_1, l_2))$, (a) the function $\min(h_l^\infty)$ is strictly decreasing, and the function $\max(h_l^\infty)$ is strictly increasing; and (b) if $\max(h_l^\infty)(\delta) = \lambda$, then $\|l_1 - l + \lambda \cdot (l_2 - l_1)\| = \delta$; and similarly for $\min(h_l^\infty)$.

For the (one dimensional) line $\text{Line}(t_{z'}, t_z)$, where $t_z \neq t_{z'}$, and a point t , we have $h_t^\infty(\delta)$ contains λ such that $|t_z - t + \lambda \cdot (t_{z'} - t_z)| \leq \delta$. This holds iff both the following hold: (a) $\lambda \cdot (t_{z'} - t_z) \leq \delta + t - t_z$; and (b) $-\lambda \cdot (t_{z'} - t_z) \leq \delta - (t - t_z)$ which is equivalent to $\lambda \cdot (t_{z'} - t_z) \geq -(\delta - (t - t_z))$. Thus,

$$h_t^\infty(\delta) = \begin{cases} \left[-\frac{\delta - (t - t_z)}{t_{z'} - t_z}, \frac{\delta + t - t_z}{t_{z'} - t_z} \right] & \text{if } (t_{z'} - t_z) > 0 \\ \left[\frac{\delta + t - t_z}{t_{z'} - t_z}, -\frac{\delta - (t - t_z)}{t_{z'} - t_z} \right] & \text{if } (t_{z'} - t_z) < 0 \end{cases} \quad (30)$$

We obtain the value of the minimum δ (denoted as δ^*) as follows. We let λ^* be the λ value when we have δ^* in Equation 29 (it can be shown that this λ^* is unique). We define the following parameters, and use them in the computation of the least δ .

- $\lambda_p^{s_1}$: this is the value, defined when $z \neq z'$, such that the point $z + \lambda_p^{s_1} \cdot (z' - z)$ on the line $\text{Line}(z, z')$ is the least distance away from s_1 in the L_2 norm. Proposition 15 gives us $\lambda_p^{s_1} = \frac{(\vec{z}' - \vec{z}) \odot (\vec{s}_1 - \vec{z})}{\|\vec{z}' - \vec{z}\|^2}$
- $\lambda_{t_{s_1}}$: this is the value, defined when $t_{z'} \neq t_z$, such that $t_{s_1} = t_z + \lambda_{t_{s_1}} \cdot (t_{z'} - t_z)$. It can be seen that $\lambda_{t_{s_1}} = \frac{t_{s_1} - t_z}{t_{z'} - t_z}$.

We also similarly have the parameters $\lambda_p^{s_2}$ and $\lambda_{t_{s_1}}$.

► Suppose $\langle s_1, t_{s_1} \rangle = \langle s_2, t_{s_2} \rangle$. Then δ^* is $\mathcal{D}_{L_2^s}(\langle s_1, t_{s_1} \rangle, \text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle))$. This can be computed using Proposition 21.

► Suppose $\langle z, t_z \rangle = \langle z', t_{z'} \rangle$. Then $\delta^* = \max \left(\|\langle s_1, t_{s_1} - \langle z, t_z \rangle \rangle\|_{L_2^s}, \|\langle s_2, t_{s_2} - \langle z, t_z \rangle \rangle\|_{L_2^s} \right)$. Thus $\delta^* = \max (\|s_1 - z\|_{L_2}, |t_{s_1} - t_z|, \|s_2 - z\|_{L_2}, |t_{s_2} - t_z|)$.

► Suppose $\langle z, t_z \rangle \neq \langle z', t_{z'} \rangle$, but $z = z'$ (and thus $t_{z'} \neq t_z$). Then,

$$\delta^* = \min_{\delta \geq 0} \left(\text{There exists } \lambda \in [0, 1] \text{ such that } \left\{ \begin{array}{l} \|z - s_1\|_{L_2} \leq \delta; \text{ and} \\ |t_z - t_{s_1} + \lambda \cdot (t_{z'} - t_z)| \leq \delta; \text{ and} \\ \|z - s_2\|_{L_2} \leq \delta; \text{ and} \\ |t_z - t_{s_2} + \lambda \cdot (t_{z'} - t_z)| \leq \delta \end{array} \right\} \right).$$

This can be simplified to:

$$\delta^* = \min_{\delta \geq \max(\|z - s_1\|_{L_2}, \|z - s_2\|_{L_2})} \left(\text{There exists } \lambda \in [0, 1] \text{ such that } \left\{ \begin{array}{l} |t_z - t_{s_1} + \lambda \cdot (t_{z'} - t_z)| \leq \delta; \text{ and} \\ |t_z - t_{s_2} + \lambda \cdot (t_{z'} - t_z)| \leq \delta \end{array} \right\} \right).$$

Using h^∞ functions, the above can be written as

$$\delta^* = \min_{\delta \geq \max(\|z - s_1\|_{L_2}, \|z - s_2\|_{L_2})} \left(h_{t_{s_1}}^\infty(\delta) \cap h_{t_{s_2}}^\infty(\delta) \cap [0, 1] \neq \emptyset \right). \quad (31)$$

The above equation is equivalent to the following:

$$\delta^\dagger(t_{s_1}, t_{s_2}, \text{Line}(t_{z'}, t_z)) = \min_{\delta \geq 0} \left(h_{t_{s_1}}^\infty(\delta) \cap h_{t_{s_2}}^\infty(\delta) \cap [0, 1] \neq \emptyset \right) \quad (32)$$

$$\delta^* = \max \left(\|z - s_1\|_{L_2}, \|z - s_2\|_{L_2}, \delta^\dagger(t_{s_1}, t_{s_2}, \text{Line}(t_{z'}, t_z)) \right) \quad (33)$$

We now show how to compute δ^\dagger (we omit the arguments $t_{s_1}, t_{s_2}, \text{Line}(t_{z'}, t_z)$ for simplicity). The intervals around $\lambda_{t_{s_1}}$ and $\lambda_{t_{s_2}}$ keep getting bigger as δ gets bigger – we want the least δ such that there is an intersection point in $[0, 1]$. Moreover, the rate at which the boundaries of the intervals increase or decrease is the same from Equation 30. We assume $\lambda_{t_{s_2}} \geq 0$ (otherwise, we swap $\langle z, t_z \rangle$ with $\langle z', t_{z'} \rangle$ to make this so). We also assume $\lambda_{t_{s_2}} \geq \lambda_{t_{s_1}}$ (this can be ensured by swapping t_{s_1} and t_{s_2} if necessary). The following cases arise (see Figure 10 for a pictorial representation of the λ value placements).

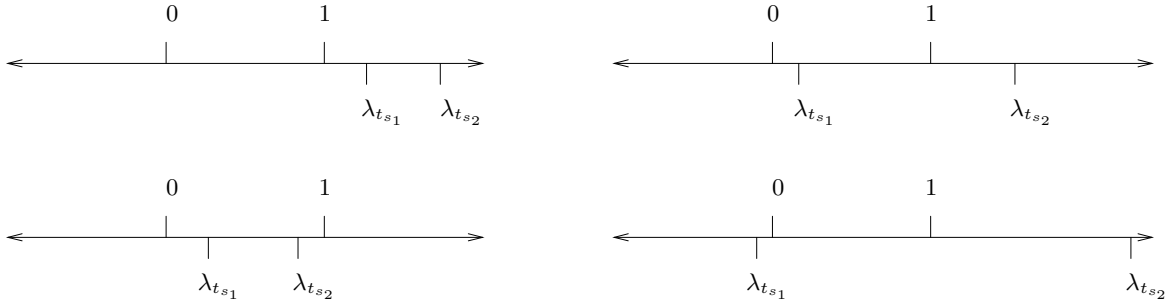


Figure 10: λ positions for $t_{z'} \neq t_z$.

1. $\lambda_{t_{s_1}}, \lambda_{t_{s_2}}$ both ≥ 1 .

Since the rate of decrease of $\min(h_{\lambda_{t_{s_1}}}^\infty)$ is the same as that of $\min(h_{\lambda_{t_{s_2}}}^\infty)$, in this case δ^\dagger is when $\min(h_{\lambda_{t_{s_2}}}^\infty)(\delta^\dagger) = 1$. That is, when $\text{Sphere}(t_{s_2}, \delta^\dagger) = t_{z'}$. Solving, we get $\delta^\dagger = |t_{s_2} - t_{z'}|$.

2. $\lambda_{t_{s_1}}, \lambda_{t_{s_2}}$ both between 0 and 1.

In this, the minimum δ is obtained when $\min(h_{\lambda_{t_{s_2}}}^\infty) = \max(h_{\lambda_{t_{s_1}}}^\infty)$. Using Equation 30, we have two cases.

(a) $(t_{z'} - t_z) > 0$. In this case $-\frac{\delta^\dagger - (t_{s_2} - t_z)}{t_{z'} - t_z} = \frac{\delta^\dagger + (t_{s_1} - t_z)}{t_{z'} - t_z}$. Simplifying, $2 \cdot \delta^\dagger = (t_{s_2} - t_z) - (t_{s_1} - t_z)$, thus, $\delta^\dagger = (t_{s_2} - t_{s_1})/2$.

(b) $(t_{z'} - t_z) < 0$. In this case $\frac{\delta^\dagger + (t_{s_2} - t_z)}{t_{z'} - t_z} = -\frac{\delta^\dagger - (t_{s_1} - t_z)}{t_{z'} - t_z}$. Solving, we get $\delta^\dagger = (t_{s_1} - t_{s_2})/2$.

Putting the two cases, together,

$$\delta^\dagger = \begin{cases} (t_{s_2} - t_{s_1})/2 & \text{if } (t_{z'} - t_z) > 0 \\ (t_{s_1} - t_{s_2})/2 & \text{if } (t_{z'} - t_z) < 0. \end{cases}$$

Or, equivalently, $\delta^\dagger = |t_{s_2} - t_{s_1}|/2$.

3. $\lambda_{t_{s_2}} > 1$, and $0 \leq \lambda_{t_{s_1}} < 1$

We need to consider two sub-case.

(a) If δ_1^1 such that $\max(h_{\lambda_{t_{s_1}}}^\infty)(\delta_1^1) = 1$ is less than, or equal to δ_2^1 such that $\min(h_{\lambda_{t_{s_2}}}^\infty)(\delta_2^1) = 1$, then, when the two intervals around $\lambda_{t_{s_1}}$ and $\lambda_{t_{s_2}}$ intersect, the intersection point is going to be in $[0, 1]$. Intuitively, the min boundary of $h_{\lambda_{t_{s_2}}}^\infty$ hits 1 before the max boundary of $h_{\lambda_{t_{s_1}}}^\infty$ does. In this case, the least δ is when the boundaries $\min(h_{\lambda_{t_{s_2}}}^\infty)$ and $\max(h_{\lambda_{t_{s_1}}}^\infty)$ intersect. The δ^\dagger value can then be extracted as in the previous case. We have $\text{Sphere}(t_{s_2}, \delta_1^1) = t_{z'}$; and $\text{Sphere}(t_{s_1}, \delta_1^1) = t_{z'}$; i.e. $\delta_1^1 = |t_{s_1} - t_{z'}|$ and $\delta_2^1 = |t_{s_2} - t_{z'}|$. Thus, if $|t_{s_1} - t_{z'}| \leq |t_{s_2} - t_{z'}|$, then $\delta^\dagger = |t_{s_2} - t_{s_1}|/2$.

(b) If δ_1^1 is greater than δ_2^1 , where these values are as defined in the previous case. In this case, $\max(h_{\lambda_{t_{s_1}}}^\infty)$ hits 1 before $\min(h_{\lambda_{t_{s_2}}}^\infty)$ does, i.e. $\max(h_{\lambda_{t_{s_1}}}^\infty)$ hits 1 before $\min(h_{\lambda_{t_{s_2}}}^\infty)$ reaches $\max(h_{\lambda_{t_{s_1}}}^\infty)$. Thus, the least δ occurs when $\min(h_{\lambda_{t_{s_2}}}^\infty)(\delta^\dagger) = 1$. That is, when $\text{Sphere}(t_{s_2}, \delta^\dagger) = t_{z'}$. Thus, $\delta^\dagger = |t_{s_2} - t_{z'}|$.

Putting the two cases together, we get

$$\delta^\dagger = \begin{cases} |t_{s_2} - t_{z'}| & \text{if } |t_{s_1} - t_{z'}| > |t_{s_2} - t_{z'}| \\ |t_{s_2} - t_{s_1}|/2 & \text{otherwise.} \end{cases}$$

4. $\lambda_{t_{s_2}} > 1$, and $\lambda_{t_{s_1}} < 0$

The analysis of this case depends on when the boundaries of $h_{\lambda_{t_{s_1}}}^\infty$ and $h_{\lambda_{t_{s_2}}}^\infty$ hit 0 and 1.

(a) Suppose the min boundary of $h_{\lambda_{t_{s_2}}}^\infty$ hits 0 before the max boundary of $h_{\lambda_{t_{s_1}}}^\infty$ hits 0. Let us call the respective δ values δ_1^0 and δ_2^0 . In this case, the least desired δ is going to be when the max boundary of $h_{\lambda_{t_{s_1}}}^\infty$ hits 0. Thus, if $|t_{s_1} - t_z| < |t_{s_2} - t_z|$, then $\delta^\dagger = |t_{s_1} - t_z|$.

(b) If the previous case does not hold, and if the min boundary of $h_{\lambda_{t_{s_2}}}^\infty$ hits 1 before the max boundary of $h_{\lambda_{t_{s_1}}}^\infty$ hits 1, then the intersection point of the two intervals in going to be in $[0, 1]$. The value can be obtained from one of the previous case. Thus, if $|t_{s_1} - t_z| \geq |t_{s_2} - t_z|$, and $|t_{s_1} - t_{z'}| \leq |t_{s_2} - t_{z'}|$, then $\delta^\dagger = |t_{s_2} - t_{s_1}|/2$.

- (c) If the previous two case do not hold, *i.e.* $|t_{s_1} - t_z| \geq |t_{s_2} - t_z|$, and $|t_{s_1} - t_{z'}| > |t_{s_2} - t_{z'}|$, then the minimum δ occurs when the min boundary of $h_{\lambda_{t_{s_2}}}^\infty$ hits 1, *i.e.* when $\delta^\dagger = |t_{s_2} - t_{z'}|$.

Putting the above three cases together,

$$\delta^\dagger(t_{s_1}, t_{s_2}, \text{Line}(t_{z'}, t_z)) = \begin{cases} |t_{s_1} - t_z| & \text{if } |t_{s_1} - t_z| < |t_{s_2} - t_z| \\ |t_{s_2} - t_{s_1}|/2 & \text{if } |t_{s_1} - t_z| \geq |t_{s_2} - t_z| \text{ and } |t_{s_1} - t_{z'}| \leq |t_{s_2} - t_{z'}| \\ |t_{s_2} - t_{z'}| & \text{otherwise.} \end{cases}$$

► Suppose $\langle z, t_z \rangle \neq \langle z', t_{z'} \rangle$, but $t_z = t_{z'}$ (and thus $z' \neq z$). Then,

$$\delta^* = \min_{\delta \geq 0} \left(\text{There exists } \lambda \in [0, 1] \text{ such that } \begin{cases} \|\vec{z} - \vec{s}_1 + \lambda \cdot (\vec{z}' - \vec{z})\|_{L_2} \leq \delta; \text{ and} \\ |t_z - t_{s_1}| \leq \delta; \text{ and} \\ \|\vec{z} - \vec{s}_2 + \lambda \cdot (\vec{z}' - \vec{z})\|_{L_2} \leq \delta; \text{ and} \\ |t_z - t_{s_2}| \leq \delta \end{cases} \right)$$

The above is equivalent to:

$$\delta^* = \min_{\delta \geq \max(|t_z - t_{s_1}|, |t_z - t_{s_2}|)} \left(\text{There exists } \lambda \in [0, 1] \text{ such that } \begin{cases} \|\vec{z} - \vec{s}_1 + \lambda \cdot (\vec{z}' - \vec{z})\|_{L_2} \leq \delta; \text{ and} \\ \|\vec{z} - \vec{s}_2 + \lambda \cdot (\vec{z}' - \vec{z})\|_{L_2} \leq \delta; \end{cases} \right)$$

The above is equivalent to

$$\delta^* = \max \left(|t_z - t_{s_1}|, |t_z - t_{s_2}|, \min_{\delta \geq 0} \{ \delta \mid \text{Ball}_{L_2}(s_1, \delta) \cap \text{Ball}_{L_2}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset \} \right)$$

This can be computed using the algorithms for the L_2 -norm.

► Suppose $\langle z, t_z \rangle \neq \langle z', t_{z'} \rangle$, and $t_z \neq t_{z'}$, and $z' \neq z$. The analysis of this case depends on the relative positions of $\lambda_p^{s_1}, \lambda_{t_{s_1}}, \lambda_p^{s_2}$, and $\lambda_{t_{s_2}}$. Figure 11 illustrates some of the λ value placements). Note that $\lambda_p^{s_1}$ and $\lambda_p^{s_2}$ belong to $[0, 1]$ by definition.

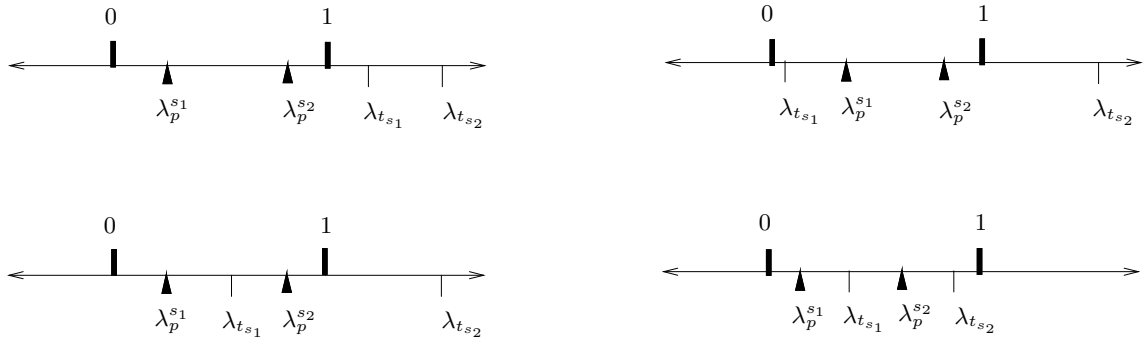


Figure 11: A few of the λ positions for $z' \neq z$ and $t_{z'} \neq t_z$.

Consider Equation 29. There are four innermost clauses. Suppose for each δ we define a set $H(\delta)$ which denote the λ points in $[0, 1]$ that satisfy all the four constraints of Equation 29 as follows:

$$H(\delta) = \left\{ \lambda \mid \begin{array}{l} \left\| \vec{z} - \vec{s}_1 + \lambda \cdot (\vec{z}' - \vec{z}) \right\|_{L_2} \leq \delta; \text{ and} \\ |t_z - t_{s_1} + \lambda \cdot (t_{z'} - t_z)| \leq \delta; \text{ and} \\ \left\| \vec{z} - \vec{s}_2 + \lambda \cdot (\vec{z}' - \vec{z}) \right\|_{L_2} \leq \delta; \text{ and} \\ |t_z - t_{s_2} + \lambda \cdot (t_{z'} - t_z)| \leq \delta; \text{ and} \\ \lambda \in [0, 1] \end{array} \right\}.$$

Then, δ^* , the optimal value of the solution of Equation 29 is:

$$\delta^* = \min_{\delta} (H(\delta) \neq \emptyset).$$

We can break down $H(\delta)$ into sets defined by individual constraints as follows:

$$\begin{aligned} H_1(\delta) &= \{\lambda \mid \left\| \vec{z} - \vec{s}_1 + \lambda \cdot (\vec{z}' - \vec{z}) \right\|_{L_2} \leq \delta\} \\ H_2(\delta) &= \{\lambda \mid |t_z - t_{s_1} + \lambda \cdot (t_{z'} - t_z)| \leq \delta\} \\ H_3(\delta) &= \{\lambda \mid \left\| \vec{z} - \vec{s}_2 + \lambda \cdot (\vec{z}' - \vec{z}) \right\|_{L_2} \leq \delta\} \\ H_4(\delta) &= \{\lambda \mid |t_z - t_{s_2} + \lambda \cdot (t_{z'} - t_z)| \leq \delta\} \\ H(\delta) &= H_1(\delta) \cap H_2(\delta) \cap H_3(\delta) \cap H_4(\delta) \cap [0, 1] \end{aligned}$$

For $1 \leq i < j \leq 4$, consider the equation defined by:

$$\delta_{i,j}^* = \min_{\delta} (H_i(\delta) \cap H_j(\delta) \cap [0, 1] \neq \emptyset)$$

It is clear that

$$\delta^* \geq \max_{1 \leq i < j \leq 4} \delta_{i,j}^*. \quad (34)$$

This is because if $\lambda \in H(\delta)$, then $\lambda \in H_i(\delta) \cap H_j(\delta) \cap [0, 1]$. We show that we in fact have equality:

$$\delta^* = \max_{1 \leq i < j \leq 4} \delta_{i,j}^*. \quad (35)$$

Observe that for any $\delta \geq \max_{1 \leq i < j \leq 4} \delta_{i,j}^*$, we have:

- $(H_i(\delta) \cap [0, 1]) \cap (H_j(\delta) \cap [0, 1]) \neq \emptyset$ for all $1 \leq i < j \leq 4$ since $\delta > \delta_{i,j}^*$.
- $H_i(\delta) \cap [0, 1]$ is an interval $[\alpha_i, \alpha'_i]$ with $\alpha'_i \geq \alpha_i$ for all $1 \leq i \leq 4$.

The second fact, which states that the set of λ values on a line such that the corresponding line points are at most δ away from some point is a closed interval, follows from Equation 30 and Lemma 5. Thus, from Helly's theorem (Proposition 22) with $d = 1$, we have that $\cap_{i=1}^4 (H_i(\delta) \cap [0, 1])$ is not empty. This means that $\delta^* \leq \max_{1 \leq i < j \leq 4} \delta_{i,j}^*$. Using Equation 34, we get $\delta^* = \max_{1 \leq i < j \leq 4} \delta_{i,j}^*$.

We now observe the following facts.

- $\min_{\delta} (H_1(\delta) \cap H_2(\delta) \cap [0, 1] \neq \emptyset)$ which equals

$$\min_{\delta \geq 0} \left(\text{There exists } \lambda \in [0, 1] \text{ such that } \left\{ \begin{array}{l} \left\| \vec{z} - \vec{s}_1 + \lambda \cdot (\vec{z}' - \vec{z}) \right\|_{L_2} \leq \delta; \text{ and} \\ |t_z - t_{s_1} + \lambda \cdot (t_{z'} - t_z)| \leq \delta \end{array} \right\} \right)$$

is equal to $\mathcal{D}_{L_2^s}(\langle s_1, t_{s_1} \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle))$ from Equation 25.

- $\min_{\delta} (H_1(\delta) \cap H_3(\delta) \cap [0, 1] \neq \emptyset)$ which equals

$$\min_{\delta \geq 0} \left(\text{There exists } \lambda \in [0, 1] \text{ such that } \left\{ \begin{array}{l} \left\| \vec{z} - \vec{s}_1 + \lambda \cdot (\vec{z}' - \vec{z}) \right\|_{L_2} \leq \delta; \text{ and } \\ \left\| \vec{z} - \vec{s}_2 + \lambda \cdot (\vec{z}' - \vec{z}) \right\|_{L_2} \leq \delta; \end{array} \right\} \right)$$

is equal to $\min_{\delta \geq 0} \{\delta \mid \text{Ball}_{L_2}(s_1, \delta) \cap \text{Ball}_{L_2}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}$.

- $\min_{\delta} (H_1(\delta) \cap H_4(\delta) \cap [0, 1] \neq \emptyset)$ which equals

$$\min_{\delta \geq 0} \left(\text{There exists } \lambda \in [0, 1] \text{ such that } \left\{ \begin{array}{l} \left\| \vec{z} - \vec{s}_1 + \lambda \cdot (\vec{z}' - \vec{z}) \right\|_{L_2} \leq \delta; \text{ and } \\ |t_z - t_{s_2} + \lambda \cdot (t_{z'} - t_z)| \leq \delta \end{array} \right\} \right)$$

is equal to $\mathcal{D}_{L_2^S}(\langle s_1, t_{s_2} \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle))$ from Equation 25.

- $\min_{\delta} (H_2(\delta) \cap H_3(\delta) \cap [0, 1] \neq \emptyset)$ which equals

$$\min_{\delta \geq 0} \left(\text{There exists } \lambda \in [0, 1] \text{ such that } \left\{ \begin{array}{l} |t_z - t_{s_1} + \lambda \cdot (t_{z'} - t_z)| \leq \delta; \text{ and } \\ \left\| \vec{z} - \vec{s}_2 + \lambda \cdot (\vec{z}' - \vec{z}) \right\|_{L_2} \leq \delta \end{array} \right\} \right)$$

which is equal to $\mathcal{D}_{L_2^S}(\langle s_2, t_{s_1} \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle))$ from Equation 25.

- $\min_{\delta} (H_2(\delta) \cap H_4(\delta) \cap [0, 1] \neq \emptyset)$ which equals

$$\min_{\delta \geq 0} \left(\text{There exists } \lambda \in [0, 1] \text{ such that } \left\{ \begin{array}{l} |t_z - t_{s_1} + \lambda \cdot (t_{z'} - t_z)| \leq \delta; \text{ and } \\ |t_z - t_{s_2} + \lambda \cdot (t_{z'} - t_z)| \leq \delta \end{array} \right\} \right)$$

is equal to $\delta^\dagger(t_{s_1}, t_{s_2}, \text{Line}(t_{z'}, t_z))$ from Equation 32.

- $\min_{\delta} (H_3(\delta) \cap H_4(\delta) \cap [0, 1] \neq \emptyset)$ which equals

$$\min_{\delta \geq 0} \left(\text{There exists } \lambda \in [0, 1] \text{ such that } \left\{ \begin{array}{l} \left\| \vec{z} - \vec{s}_2 + \lambda \cdot (\vec{z}' - \vec{z}) \right\|_{L_2} \leq \delta; \text{ and } \\ |t_z - t_{s_2} + \lambda \cdot (t_{z'} - t_z)| \leq \delta \end{array} \right\} \right)$$

is equal to $\mathcal{D}_{L_2^S}(\langle s_2, t_{s_2} \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle))$ from Equation 25.

Thus, in case $\langle z, t_z \rangle \neq \langle z', t_{z'} \rangle$, and $t_z \neq t_{z'}$, and $z' \neq z$, putting everything together, we have:

$$\delta^* = \max \left(\begin{array}{c} \mathcal{D}_{L_2^S}(\langle s_1, t_{s_1} \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle)), \\ \min_{\delta \geq 0} \{\delta \mid \text{Ball}_{L_2}(s_1, \delta) \cap \text{Ball}_{L_2}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset\}, \\ \mathcal{D}_{L_2^S}(\langle s_1, t_{s_2} \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle)), \\ \mathcal{D}_{L_2^S}(\langle s_2, t_{s_1} \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle)), \\ \delta^\dagger(t_{s_1}, t_{s_2}, \text{Line}(t_{z'}, t_z)), \\ \mathcal{D}_{L_2^S}(\langle s_2, t_{s_2} \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle)) \end{array} \right).$$

We note that we have shown how to compute all the individual values inside the max set before.

The following function $\Phi_{L_2^S}$ combines everything together giving us Proposition 23.

Proposition 23 (Computation of least δ such that $\text{Ball}(\langle s_1, t_{s_1} \rangle, \delta) \cap \text{Ball}(\langle s_2, t_{s_1} \rangle, \delta) \cap \text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle)$ is non-empty: L_2^S). *Given points $\langle s_1, t_{s_1} \rangle, \langle s_2, t_{s_2} \rangle, \langle z, t_z \rangle, \langle z', t_{z'} \rangle$ with $s_1, s_2, z, z' \in \mathbb{R}^n$ and $t_{s_1}, t_{s_2}, t_z, t_{z'} \in \mathbb{R}$. Function $\Phi_{L_2^S}$ computes*

$$\min_{\delta \geq 0} \{\delta \mid \text{Ball}(\langle s_1, t_{s_1} \rangle, \delta) \cap \text{Ball}(\langle s_2, t_{s_2} \rangle, \delta) \cap \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle) \neq \emptyset\}$$

for the L_2^S norm. □


```

Input : Points  $\langle s_1, t_{s_1} \rangle, \langle s_1, t_{s_2} \rangle, \langle z, t_z \rangle, \langle z', t_{z'} \rangle$  in  $\mathbb{R}^n \times \mathbb{R}$ 
Output: Least  $\delta$  such that  $\text{Ball}(\langle s_1, t_{s_1} \rangle, \delta) \cap \text{Ball}(\langle s_2, t_{s_1} \rangle, \delta) \cap \text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle)$  is
non-empty in  $L_2^S$  norm

if  $t_{z'} \neq t_z$  then                                     // ensure  $\lambda_{t_{s_2}} \geq 0$  and  $\lambda_{t_{s_2}} \geq \lambda_{t_{s_1}}$ 
     $\lambda_{t_{s_1}} = \frac{t_{s_1} - t_z}{t_{z'} - t_z};$ 
     $\lambda_{t_{s_2}} = \frac{t_{s_2} - t_z}{t_{z'} - t_z};$ 
    if  $\lambda_{t_{s_2}} < 0$  then
        swap  $\langle z, t_z \rangle$  with  $\langle z', t_{z'} \rangle;$ 
         $\lambda_{t_{s_1}} = 1 - \lambda_{t_{s_1}};$ 
         $\lambda_{t_{s_2}} = 1 - \lambda_{t_{s_2}};$ 
    end
    if  $\lambda_{t_{s_2}} < \lambda_{t_{s_1}}$  then
        swap  $t_{s_2}$  with  $t_{s_1};$ 
        swap  $\lambda_{t_{s_2}}$  with  $\lambda_{t_{s_1}};$ 
    end
end

if  $t_{z'} \neq t_z$  then                                     // we need  $\delta^\dagger$  twice
     $\delta^\dagger := \begin{cases} |t_{s_1} - t_z| & \text{if } |t_{s_1} - t_z| < |t_{s_2} - t_z| \\ |t_{s_2} - t_{s_1}|/2 & \text{if } |t_{s_1} - t_z| \geq |t_{s_2} - t_z| \text{ and } |t_{s_1} - t_{z'}| \leq |t_{s_2} - t_{z'}|; \\ |t_{s_2} - t_{z'}| & \text{otherwise.} \end{cases}$ 
end

switch  $\langle s_1, t_{s_1} \rangle, \langle s_1, t_{s_2} \rangle, \langle z, t_z \rangle, \langle z', t_{z'} \rangle$  do
    case  $\langle s_1, t_{s_1} \rangle = \langle s_1, t_{s_2} \rangle$  return  $\mathcal{D}_{L_2^S}(\langle s_1, t_{s_1} \rangle, \text{Line}(\langle z, t_z \rangle, \langle z', t_{z'} \rangle));$ 
    case  $\langle z, t_z \rangle = \langle z', t_{z'} \rangle$  return  $\max(|s_1 - z|_{L_2}, |t_{s_1} - t_z|, \|s_2 - z\|_{L_2}, |t_{s_2} - t_z|);$ 
    case  $z = z'$  and  $t_{z'} \neq t_z$  return  $\max(\|z - s_1\|_{L_2}, \|z - s_2\|_{L_2}, \delta^\dagger);$ 
    case  $z \neq z'$  and  $t_{z'} = t_z$ 
         $\delta_{L_2} := \min_{\delta \geq 0} \{ \delta \mid \text{Ball}_{L_2}(s_1, \delta) \cap \text{Ball}_{L_2}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset \};$ 
        //  $\delta_{L_2}$  can be computed using procedure for  $L_2$ 
        return  $\max(|t_z - t_{s_1}|, |t_z - t_{s_2}|, \delta_{L_2});$ 
    case  $z \neq z'$  and  $t_{z'} \neq t_z$ 
         $\delta_{L_2} := \min_{\delta \geq 0} \{ \delta \mid \text{Ball}_{L_2}(s_1, \delta) \cap \text{Ball}_{L_2}(s_2, \delta) \cap \text{Line}(z, z') \neq \emptyset \};$ 
        return  $\max \left( \begin{array}{c} \mathcal{D}_{L_2^S}(\langle s_1, t_{s_1} \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle)), \\ \delta_{L_2} \\ \mathcal{D}_{L_2^S}(\langle s_1, t_{s_2} \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle)), \\ \mathcal{D}_{L_2^S}(\langle s_2, t_{s_1} \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle)), \\ \delta^\dagger, \\ \mathcal{D}_{L_2^S}(\langle s_2, t_{s_2} \rangle, \text{Line}(\langle z, t_z \rangle, \langle z, t_{z'} \rangle)) \end{array} \right);$ 
    endsw

```

Function $\Phi_{L_2^S}(\langle s_1, t_{s_1} \rangle, \langle s_1, t_{s_2} \rangle, \langle z, t_z \rangle, \langle z', t_{z'} \rangle)$

6 The Skorokhod Distance Algorithm

Using the reduction from Proposition 1 and the algorithm from Theorem 1, together with the procedures from Propositions 19, 20, 21, 23, 17 and 18 for computing geometric primitives, we obtain the complete algorithm for computing the continuous time Skorokhod distance between two polygonal traces for the L_1, L_2 and L_∞ norms.

Theorem 2. *Let $x : [0, T_x] \mapsto \mathbb{R}^n$ and $y : [0, T_y] \mapsto \mathbb{R}^n$ be two polygonal traces with m_x and m_y affine segments respectively. The continuous time Skorokhod distance between them, denoted $\mathcal{S}_\chi(x, y)$ for the norm $\chi \in \{L_1, L_2, L_\infty\}$ can be computed in time:*

$$O\left((m_y \cdot m_x^2 + m_x \cdot m_y^2) \cdot (P(\chi) + \log(m_y \cdot m_x)) + m_x \cdot m_y \cdot H(\chi)\right)$$

where

- for L_1 , we have $P(L_1) = \text{LP}(n)$ and $H(L_1) = n^2$.
- for L_2 , we have $P(L_2) = n$ and $H(L_2) = n$.
- for L_∞ , we have $P(L_\infty) = \text{LP}(n)$ and $H(L_\infty) = n$.

and where $\text{LP}(n)$ is the (polynomial-time) upper bound for linear programming. The corresponding decision problem can be solved in time $O(m_x \cdot m_y \cdot H(\chi))$. \square

The Skorokhod Distance with Windows. As for the Fréchet distance, it is often of interest to apply a window of W in the the retimings. The complexity of computing the value of the Skorokhod distance in this case is $O\left(W^2 \cdot M \cdot (P(\chi) + \log(W \cdot M)) + W \cdot M \cdot \log(W \cdot M) \cdot H(\chi)\right)$, where $M = \max(m_f, m_g)$; and $H(\chi)$, and $P(\chi)$ are as in Theorem 2. The corresponding decision problem runs in time $O(M \cdot W^2 \cdot H(\chi))$. If W can be taken to be a constant, the value computation algorithm takes $O\left(M \cdot P(\chi) + M \cdot \log(M) \cdot H(\chi)\right)$ time, and the decision problem can be solved in $O(M \cdot H(\chi))$ time.

7 Conclusion

Our work presents the first algorithm for computing the Skorokhod distance between polygonal traces in \mathbb{R}^n ; such traces arise when sampled-time traces are completed by linear interpolation. The individual dimensional values, and also the time, can be scaled by different constants to suit the needs of a particular problem. Our algorithm is fully polynomial time for the three norms L_1, L_2 and L_∞ , and runs in $O(m^3 \cdot \log(m) \cdot \text{poly}(n))$ time, where values come from \mathbb{R}^n and m is the number of affine segments.⁵ The polynomial factor $\text{poly}(n)$ depends on the norm, and represents the cost of computing the two geometric primitives. For example, for the L_1, L_∞ , and L_1^S norms, the computation involves linear programming, and for L_2 and L_2^S , it requires solving multiple quadratic equations. Synergistically, our work also provides complete (and fully polynomial-time) algorithms for computing the Fréchet distance between curves in \mathbb{R}^n for the five norms.

In many practical applications, one can restrict attention to retimings within a “sliding window” of constant size (e.g., we may want to match a point in one trace to a point at most 5 units away

⁵The algorithm can be improved to run in $O(m^2 \cdot \log(m) \cdot \text{poly}(n))$ time using parametric search and parallel sorting, as mentioned in [AG95] for Fréchet metrics in \mathbb{R}^2 ; however these improvements (which can be added on top of the work presented) are not easily implementable and often involve huge constants.

in the other trace). We present a window optimization that can be used to gain further efficiency (the decision problem can then be solved in linear time assuming a constant dimension for \mathbb{R}^n).

While we consider the max norm modulo retimings in this work, an interesting question is to design polynomial time algorithms for norms that involve summing the differences of two traces after retiming.

References

- [AG95] H. Alt and M. Godau. Computing the fr chet distance between two polygonal curves. *Int. J. Comput. Geometry Appl.*, 5:75–91, 1995.
- [AHF⁺14] H. Abbas, B. Hoxha, G.E. Fainekos, J.V. Deshmukh, J. Kapinski, and K. Ueda. Conformance testing as falsification for cyber-physical systems. In *ICCPs 2014*, page 211. IEEE, 2014.
- [AMF14] H. Abbas, H. Mittelman, and G. Fainekos. Formal property verification in a conformance testing framework. In *MEMOCODE*. IEEE, 2014.
- [BBW08] K. Buchin, M. Buchin, and C. Wenk. Computing the fr chet distance between simple polygons. *Comput. Geom.*, 41(1-2):2–20, 2008.
- [BC96] D. J. Berndt and J. Clifford. Finding patterns in time series: A dynamic programming approach. In *Advances in Knowledge Discovery and Data Mining*, pages 229–248. American Association for Artificial Intelligence, 1996.
- [Blo79] D. M. Bloom. *Linear Algebra and Geometry*. Cambridge University Press, 1979.
- [BR06] J. Bisschop and M. Roelofs. *AIMMS optimization modeling*. Paragon Decision Technology, 2006.
- [Bro98] M. Broucke. Regularity of solutions and homotopic equivalence for hybrid systems. In *IEEE Conference on Decision and Control*, volume 4, pages 4283–4288, Dec 1998.
- [CB02] P. Caspi and A. Benveniste. Toward an approximation theory for computerised control. In *EMSOFT*, pages 294–304. Springer, 2002.
- [CdVE⁺10] E. W. Chambers, E. C. de Verdi re, J. Erickson, S. Lazard, F. Lazarus, and S. Thite. Homotopic Fr chet distance between curves or, walking your dog in the woods in polynomial time. *Comput. Geom. Theory Appl.*, 43(3):295–311, April 2010.
- [CIM14] K. Chatterjee, R. Ibsen-Jensen, and R. Majumdar. Edit distance for timed automata. In *HSCC 14*, pages 303–312. ACM, 2014.
- [CP13] K. Chatterjee and V. S. Prabhu. Quantitative timed simulation functions and refinement metrics for real-time systems. In *HSCC*. ACM, 2013.
- [DEG12] D.A.Branna, M. F. Esplen, and J. J. Gray. *Geometry*. Cambridge University Press, 2012.

- [Eck93] J. Eckhoff. Helly, radon, and carathéodory type theorems. In *Handbook of Convex Geometry, Vol A*, pages 389–448. North-Holland, 1993.
- [EVF07] A. Efrat, S. Venkatasubramanian, and Q. Fan. Curve matching, time warping, and light fields: New algorithms for computing similarity between curves. *J. Mathematic Imaging and Vision*, 2007.
- [FSUY12] G. Fainekos, S. Sankaranarayanan, K. Ueda, and H. Yazarel. Verification of automotive control applications using S-TaLiRo. In *ACC 12*, 2012.
- [Kna02] C. Knauer. *Algorithms for Comparing Geometric Patterns*. PhD thesis, Fachbereich Mathematik u. Informatik, Freie Universität Berlin, 2002.
- [MSSZ11] A. Maheshwari, J. R. Sack, K. Shahbaz, and H. Z. Zadeh. Fréchet distance with speed limits. *Comput. Geom.*, 44(2):110–120, 2011.
- [Mül07] M. Müller. Dynamic time warping. In *Information Retrieval for Music and Motion*, pages 69–84. Springer Berlin Heidelberg, 2007.
- [SC07] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580, October 2007.
- [Sko56] A. Skorokhod. Limit theorems for stochastic processes. *Theory of Probability and Its Applications*, 1(3):261–290, 1956.
- [Slo01] D. Sloughter. *The Calculus of Functions of Several Variables*. D. Sloughter, 2001.
- [Tab09] P. Tabuada. *Verification and Control of Hybrid Systems: A symbolic approach*. Springer, 2009.
- [Wen03] C. Wenk. *Shape Matching in Higher Dimensions*. PhD thesis, Fachbereich Mathematik u. Informatik, Freie Universität Berlin, 2003.

8 Appendix

We show how to compute the parameters $\ddot{\mathbf{a}}_{i,j}^q, \ddot{\mathbf{b}}_{i,j}^q$ for $0 \leq q \leq 3$, and $\text{cpoint}_{i,j}$ as follows. We use the convention that when we say a coordinate of either $\ddot{\mathbf{a}}_{i,j}^q, \ddot{\mathbf{b}}_{i,j}^q$ has the value $d \in \mathbb{R}$ we mean the value $\langle d, \cdot \rangle$; we also let \bar{d} stand for either $\langle d, \cdot \rangle$ or $\langle d, + \rangle$. We also define $\langle d, \psi \rangle + \langle 0, + \rangle = \langle d, + \rangle$ for $\psi \in \{\cdot, +\}$. The value \perp added to anything is \perp . We use the expected ordering for $\{\cdot, +\}$; namely $\langle d, \cdot \rangle < \langle d, + \rangle$. We observe that for $i > 0, j > 0$, there exists a monotone curve from $(0, 0)$ to the free space boundaries of cell i, j iff one of the following three hold:

1. There exists a free space monotone curve to the right edge of cell $i - 1, j$ which can continue onto the free space of cell i, j . See Figure 12 Note that if $\text{Line}(\ddot{\mathbf{a}}_{i-1,j}^1, \ddot{\mathbf{b}}_{i-1,j}^1)$ is non-empty then $\ddot{\mathbf{b}}_{i-1,j}^1 = \mathbf{b}_{i,j}^3$. This is because any strict monotone increasing curve from $(0, 0)$ that can reach $\ddot{\mathbf{a}}_{i-1,j}^1$, passing through the interior of cell $i - 1, j$, can be modified in cell $i - 1, j$ to reach $\mathbf{b}_{i-1,j}^1 (= \mathbf{b}_{i,j}^3)$.

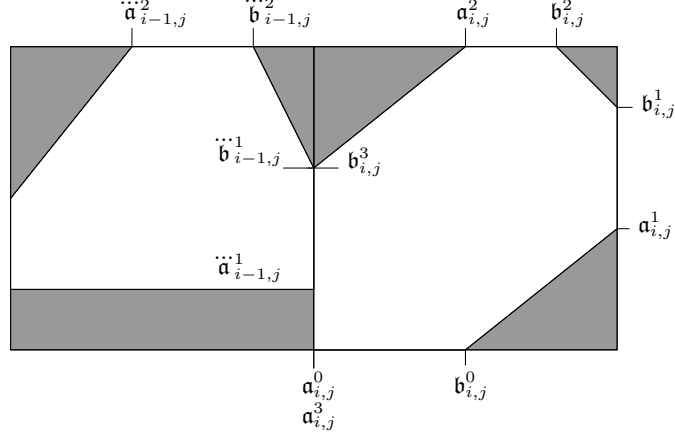


Figure 12: Cell i, j and $i - 1, j$ in $\text{Free}_\delta(f, g)$.

2. There exists a free space monotone curve to the top edge of cell $i, j - 1$ which can continue onto the free space of cell i, j . See Figure 13. We observe that if $\text{Line}(\ddot{a}_{i,j-1}^2, \ddot{b}_{i,j-1}^2)$ is non-empty then $\ddot{b}_{i,j-1}^2 = \ddot{b}_{i,j}^0$. This is because any strict monotone increasing curve from $(0, 0)$ that can reach $\ddot{a}_{i,j-1}^2$, passing through the interior of cell $i, j - 1$, can be modified in cell $i, j - 1$ to reach $\ddot{b}_{i,j-1}^2 (= \ddot{b}_{i,j}^0)$.
3. There exists a free space monotone curve to the point i, j of cell $i - 1, j - 1$ which can continue onto the free space of cell i, j . See Figure 14. However, this case can be subsumed in the previous two, as the point i, j of cell $i - 1, j - 1$ is also a point of cell $i - 1, j$, i.e., the cell to the left of cell i, j .

Note that in contrast to the free space, the portion of the cell reachable by a strict monotone increasing curve from $(0, 0)$ need not be convex. Based on the above analysis, we define monotone curve reachable free space cell boundaries as follows recursively. In the following, we assume that the left, or bottom free space cell boundaries of cell i, j are non-empty. If they both are empty, then the $\ddot{a}_{i,j}^q, \ddot{b}_{i,j}^q$ parameters get the value \perp , as in this case no monotone increasing curve can enter cell i, j . Moreover, for all q , if $\ddot{a}_{i,j}^q = \perp$ then $\ddot{a}_{i,j}^q = \perp$; and if $\ddot{b}_{i,j}^q = \perp$ then $\ddot{b}_{i,j}^q = \perp$.

Boundary Point. Intuitively, this set contains the left and bottommost boundary point if it is reachable from a strict monotone curve. We define $\text{cpoint}_{0,0} = \{(0, 0)\}$. For $i > 0, j > 0$, we define $\text{cpoint}_{i,0} = \text{cpoint}_{0,j} = \emptyset$. Note that $(i, 0)$ and $(0, j)$ for $i > 0, j > 0$ are not reachable by a monotone curve from $(0, 0)$ since both coordinates must keep *strictly* increasing. For $i > 0, j > 0$ we define $\text{cpoint}_{i,j} = \{(i, j)\}$ iff $\text{first}(\ddot{b}_{i-1,j-1}^2) = i$ or $\text{second}(\ddot{b}_{i-1,j-1}^1) = j$. See Figure 14 which illustrates the case when $\text{cpoint}_{i,j} = \{(i, j)\}$.

Left Subline. We define $\ddot{a}_{0,0}^3 = \ddot{b}_{0,0}^3 = 0$ if $(0, 0) \in \text{Free}_\delta(f, g)$; otherwise these values are \perp . For all $j > 0$ we define $\ddot{a}_{0,j}^3 = \ddot{b}_{0,j}^3 = \perp$. For $i > 0, j \geq 0$, we define $\ddot{a}_{i,j}^3 = \ddot{a}_{i-1,j}^1$, and $\ddot{b}_{i,j}^3 = \ddot{b}_{i-1,j}^1$. That is, we reduce the computation of the reachable portion of the left boundary of cell i, j to the computation of the reachable portion of the right boundary of cell $i - 1, j$. See Figure 12 which illustrates this case.

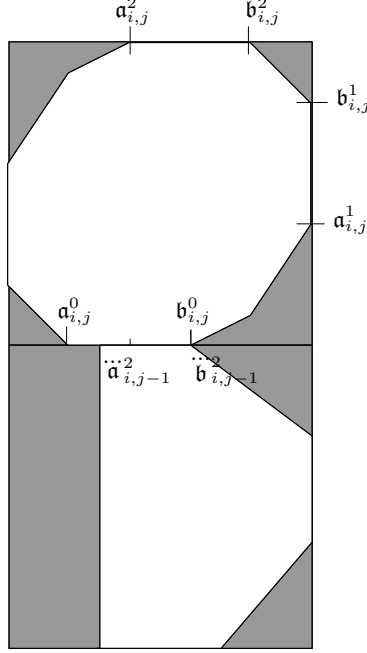


Figure 13: Cell i, j and $i, j - 1$ in $\text{Free}_\delta(f, g)$.

Bottom Subline. We define $\ddot{a}_{0,0}^0 = \ddot{b}_{0,0}^0 = 0$ if $(0,0) \in \text{Free}_\delta(f, g)$; otherwise these values are \perp . We define $\ddot{a}_{i,0}^0 = \ddot{b}_{i,0}^0 = \perp$ for all $i > 0$. For $i \geq 0, j > 0$, we define $\ddot{a}_{i,j}^0 = \ddot{a}_{i,j-1}^2$, and $\ddot{b}_{i,j}^0 = \ddot{b}_{i,j-1}^2$. That is, we reduce the computation of the reachable portion of the bottom boundary of cell i, j to the computation of the reachable portion of the upper boundary of cell $i, j - 1$.

See Figure 13 which illustrates this case.

Right Subline. We observe the following facts.

1. For cell i, j , if a point (d, j) with $d < i + 1$ on the bottom boundary of the cell i, j is reachable by a free space monotone curve, then all points on $\text{Line}(\mathbf{a}_{i,j}^1, \mathbf{b}_{i,j}^1)$ are reachable by a free space monotone continuation of that curve.
2. For cell i, j , let a point p be reachable by a free space monotone curve passing through the left boundary point (i, d) . Then any other free space monotone curve passing through a lower left boundary point (i, d') with $d' < d$ can be extended to be a free space monotone curve to p . Thus, the lower the point on the left boundary, the “better” it is for a monotone curve to reach the most points on the right boundary.

Using the above facts, we obtain the parameters $\ddot{a}_{i,j}^1, \ddot{b}_{i,j}^1$ as follows. We assume $(0,0)$ belongs to the free space, if not, then all parameters are \perp .

1. We define $\ddot{a}_{0,0}^1 = \mathbf{a}_{0,0}^1$ if $\text{second}(\mathbf{a}_{0,0}^1) > 0$, otherwise $\ddot{a}_{0,0}^1 = (1, \langle 0, + \rangle)$. Similarly, we also define $\ddot{b}_{0,0}^1 = \mathbf{b}_{0,0}^1$ if $\text{second}(\mathbf{b}_{0,0}^1) > 0$, otherwise $\ddot{b}_{0,0}^1 = (1, \langle 0, + \rangle)$. The reason is that any point on the line segment $\text{Line}(\mathbf{a}_{0,0}^1, \mathbf{b}_{0,0}^1)$ can be reached from $(0,0)$ by a strict monotone curve, except the point $(1,0)$.

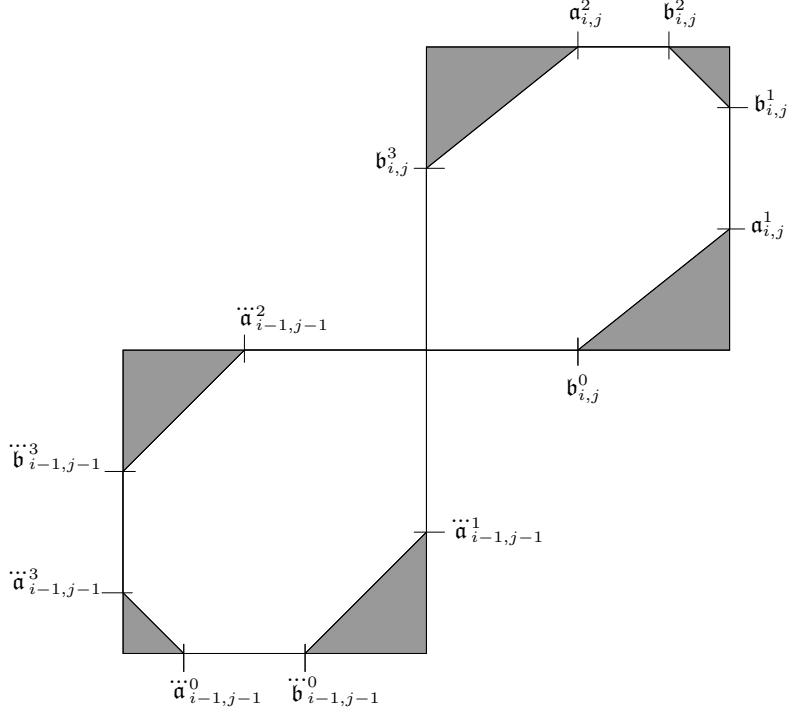


Figure 14: Cell i, j and $i - 1, j - 1$ in $\text{Free}_\delta(f, g)$.

2. For $i > 0, j = 0$ (i.e., the first row), we define $\ddot{a}_{i,0}^1$ and $\ddot{b}_{i,0}^1$ as follows using $\ddot{a}_{i,0}^3$ and $\ddot{a}_{i,0}^3$; thus, we obtain the parameters for the right boundary using the parameters for the left boundary of the same cell. Note that in this case a monotone curve can enter the free space of cell $i, 0$ only from the cell $i - 1, 0$ (the cell to the left).
 - If $\text{second}(\ddot{a}_{i,0}^3) \geq \text{second}(\ddot{b}_{i,0}^1)$, then $\ddot{a}_{i,0}^1 = \ddot{b}_{i,0}^1 = \perp$ as in this case a strictly increasing monotone curve cannot reach the right boundary of the cell $i, 0$ at all.
 - Otherwise if $\text{second}(\ddot{a}_{i,0}^3) < \text{second}(\ddot{b}_{i,0}^1)$,
 - * If $\text{second}(\ddot{a}_{i,0}^3) = \text{second}(\ddot{a}_{i,0}^1)$, then $\ddot{a}_{i,0}^1 = \ddot{a}_{i,0}^1 + \langle +, 0 \rangle$ (we add a $+$ as the point $\ddot{a}_{i,0}^1$ itself is unreachable by a strict monotone curve); and $\ddot{b}_{i,0}^1 = \ddot{b}_{i,0}^1$. If after this, we get $\ddot{a}_{i,0}^1 > \ddot{b}_{i,0}^1$, we set both to \perp .
 - * Otherwise if $\text{second}(\ddot{a}_{i,0}^3) < \text{second}(\ddot{a}_{i,0}^1)$, then $\ddot{a}_{i,0}^1 = \ddot{a}_{i,0}^1$ and $\ddot{b}_{i,0}^1 = \ddot{b}_{i,0}^1$.
3. For $0, j > 0$ (the first column), we define $\ddot{a}_{0,j}^1$ and $\ddot{b}_{0,j}^1$ as follows. Note that in this case a monotone curve can enter the free space of cell $0, j$ only from the top end of cell $0, j - 1$. If $\text{Line}(\ddot{a}_{0,j}^0, \ddot{b}_{0,j}^0)$ is empty, then both $\ddot{a}_{0,j}^1$ and $\ddot{b}_{0,j}^1$ have the value \perp . Otherwise,
 - If $\text{first}(\ddot{a}_{0,j}^0) = \langle 1, \cdot \rangle$, (in this case we also have $\text{first}(\ddot{b}_{0,j}^0) = \langle 1, \cdot \rangle$), then $\ddot{a}_{0,j}^1 = \ddot{b}_{0,j}^1 = \langle 1, j, \cdot \rangle$, as the bottom point of the right edge is the only point allowed by a strictly increasing monotone curve.
 - $\text{first}(\ddot{a}_{0,j}^0) < \langle 1, \cdot \rangle$, then, $\ddot{a}_{0,j}^1 = \ddot{a}_{0,j}^1$ and $\ddot{b}_{0,j}^1 = \ddot{b}_{0,j}^1$. This is because all point on the right boundary $\text{Line}(\ddot{a}_{0,j}^1, \ddot{b}_{0,j}^1)$ can be reached from the point on the bottom boundary $\ddot{a}_{0,j}^0$.

4. For $i > 0, j > 0$,
 - if $\text{cpoint}_{i,j} = \{(i, j)\}$, we have $\ddot{\mathbf{a}}_{i,j}^1 = \max(\mathbf{a}_{i,j}^1, (i+1, \langle j, + \rangle))$, and $\ddot{\mathbf{b}}_{i,j}^1 = \max(\mathbf{b}_{i,j}^1, (i+1, \langle j, + \rangle))$. This is because the entire line segment $\text{Line}(\mathbf{a}_{i,j}^1, \mathbf{b}_{i,j}^1)$ except for the point $(i+1, \langle j, \cdot \rangle)$ can be reached by a strictly increasing monotone curve from the corner point (i, j) .
 - if $\text{Line}(\ddot{\mathbf{a}}_{i,j}^0, \ddot{\mathbf{b}}_{i,j}^0)$ is non-empty and $\text{cpoint}_{i,j} = \emptyset$:
 - * If $\text{first}(\ddot{\mathbf{a}}_{i,j}^0) < \langle i+1, \cdot \rangle$, then $\ddot{\mathbf{a}}_{i,j}^1 = \mathbf{a}_{i,j}^1$ and $\ddot{\mathbf{b}}_{i,j}^1 = \mathbf{b}_{i,j}^1$. This corresponds to the case where the free space monotone curve enters cell i, j through either the corner point i, j , or through the bottom edge (through a point which is not $(i+1, j)$ of the cell).
 - * If $\text{first}(\ddot{\mathbf{a}}_{i,j}^0) = \langle i+1, \cdot \rangle$ and $\text{cpoint}_{i,j} = \emptyset$, then $\ddot{\mathbf{a}}_{i,j}^1 = \ddot{\mathbf{b}}_{i,j}^1 = (i+1, \langle j, \cdot \rangle)$. *initially*. We change these values if step \dagger below adds any more reachable points on the right boundary. The point $(i+1, j)$ will be covered by the corner point of cell $i+1, j$ ⁶.
 - Otherwise ($\text{Line}(\ddot{\mathbf{a}}_{i,j}^0, \ddot{\mathbf{b}}_{i,j}^0)$ and $\text{cpoint}_{i,j}$ are empty, and the analysis is as in the case $i > 0, j = 0$ done previously. (the monotone increasing curve enters the cell at the left edge):
 - * If $\text{second}(\ddot{\mathbf{a}}_{i,j}^3) \geq \text{second}(\mathbf{b}_{i,j}^1)$, then $\ddot{\mathbf{a}}_{i,j}^1 = \ddot{\mathbf{b}}_{i,j}^1 = \perp$ as in this case a strictly increasing monotone curve cannot reach the right boundary of the cell i, j at all.
 - * Otherwise if $\text{second}(\ddot{\mathbf{a}}_{i,j}^3) < \text{second}(\mathbf{b}_{i,j}^1)$, (**Step \dagger**)
 - If $\text{second}(\ddot{\mathbf{a}}_{i,j}^3) = \text{second}(\mathbf{a}_{i,j}^1)$, then $\ddot{\mathbf{a}}_{i,j}^1 = \mathbf{a}_{i,j}^1 + \langle +, 0 \rangle$ (we add a $+$ as the point $\mathbf{a}_{i,j}^1$ itself is unreachable by a strict monotone curve); and $\ddot{\mathbf{b}}_{i,j}^1 = \mathbf{b}_{i,j}^1$. If after this, we get $\ddot{\mathbf{a}}_{i,j}^1 > \ddot{\mathbf{b}}_{i,j}^1$, we set both to \perp .
 - Otherwise if $\text{second}(\ddot{\mathbf{a}}_{i,j}^3) < \text{second}(\mathbf{a}_{i,j}^1)$, then $\ddot{\mathbf{a}}_{i,j}^1 = \mathbf{a}_{i,0}^1$ and $\ddot{\mathbf{b}}_{i,j}^1 = \mathbf{b}_{i,j}^1$.

Top Subline. The analysis of this case is similar to that for the right subline. We observe the following facts.

1. For cell i, j , if a point (i, d) with $d < j+1$ on the left boundary of the cell is reachable by a free space monotone curve, then all points on $\text{Line}(\mathbf{a}_{i,j}^2, \mathbf{b}_{i,j}^2)$ are reachable by a free space monotone continuation of that curve.
2. For cell i, j , let a point p be reachable by a free space monotone curve passing through the bottom boundary point (d, j) . Then any other free space monotone curve passing through a “more left” bottom boundary point (d', j) with $d' < d$ can be extended to be a free space monotone curve to p . Thus, the more left the point on the bottom boundary, the “better” it is for a monotone curve to reach the most points on the top boundary.

Using the above facts, we obtain the parameters $\ddot{\mathbf{a}}_{i,j}^2, \ddot{\mathbf{b}}_{i,j}^2$ as follows.

1. We define $\ddot{\mathbf{a}}_{0,0}^2 = \mathbf{a}_{0,0}^2$ if $\mathbf{a}_{0,0}^2 > (0, 1)$, otherwise $\ddot{\mathbf{a}}_{0,0}^1 = (\langle 0, + \rangle, \langle 1, \cdot \rangle)$. Similarly, we also define $\ddot{\mathbf{b}}_{0,0}^2 = \mathbf{b}_{0,0}^2$ if $\mathbf{b}_{0,0}^2 > (0, 1)$, otherwise $\ddot{\mathbf{b}}_{0,0}^1 = (\langle 0, + \rangle, \langle 1, \cdot \rangle)$. The reason is that any point on the line segment $\text{Line}(\mathbf{a}_{0,0}^2, \mathbf{b}_{0,0}^2)$ can be reached from $(0, 0)$ by a strict monotone curve, except the point $(0, 1)$.

⁶Here we see the utility of keeping track of the corner points. If we required only non-decreasing monotone curves, then the whole segment $\text{Line}(\mathbf{a}_{i,j}^1, \mathbf{b}_{i,j}^1)$ would be reachable from the corner point $(i+1, j)$, and there would not be a need to treat corner points differently.

2. For $i = 0, j > 0$, we define $\ddot{\mathbf{a}}_{i,0}^1$ and $\ddot{\mathbf{b}}_{i,0}^1$ as follows. Note that in this case a monotone curve can enter the free space of cell $i, 0$ only from the cell $0, j - 1$.
 - If $\ddot{\mathbf{a}}_{0,j}^0 \geq \mathbf{b}_{0,j}^2$, then $\ddot{\mathbf{a}}_{0,j}^2 = \ddot{\mathbf{b}}_{0,j}^2 = \perp$ as in this case a strictly increasing monotone curve cannot reach the top boundary of the cell $i, 0$.
 - Otherwise if $\ddot{\mathbf{a}}_{0,j}^0 = \mathbf{a}_{0,j}^2$, then (and $\ddot{\mathbf{a}}_{0,j}^0 < \mathbf{b}_{0,j}^2$), then $\ddot{\mathbf{a}}_{0,j}^2 = \mathbf{a}_{0,j}^2 + \langle 0, + \rangle$ (we add a $+$ as the point $\mathbf{a}_{0,j}^2$ itself is unreachable by a strict monotone curve); and $\ddot{\mathbf{b}}_{0,j}^2 = \mathbf{b}_{0,j}^2$.
 - Otherwise if $\ddot{\mathbf{a}}_{0,j}^0 \neq \mathbf{a}_{0,j}^2$, (and the previous two cases do not hold), then $\ddot{\mathbf{a}}_{0,j}^2 = \max(\ddot{\mathbf{a}}_{0,j}^0, \mathbf{a}_{0,j}^2)$; and $\ddot{\mathbf{b}}_{0,j}^2 = \mathbf{b}_{0,j}^2$.
3. For $i > 0, j = 0$, we define $\ddot{\mathbf{a}}_{i,0}^2$ and $\ddot{\mathbf{b}}_{i,0}^2$ as follows. Note that in this case a monotone curve can enter the free space of cell $i, 0$ only from the right end of cell $i - 1, 0$. If $\text{Line}(\ddot{\mathbf{a}}_{i,0}^3, \ddot{\mathbf{b}}_{i,0}^3)$ is non-empty, then $\ddot{\mathbf{a}}_{i,0}^2 = \mathbf{a}_{i,0}^2$ and $\ddot{\mathbf{b}}_{i,0}^2 = \mathbf{b}_{i,0}^2$; otherwise both $\ddot{\mathbf{a}}_{i,0}^2$ and $\ddot{\mathbf{b}}_{i,0}^2$ have the value \perp .
4. For $i > 0, j > 0$,
 - if either $\text{Line}(\ddot{\mathbf{a}}_{i,j}^3, \ddot{\mathbf{b}}_{i,j}^3)$ is non-empty, or $\text{cpoint}_{i,j} = \{(i, j)\}$, then $\ddot{\mathbf{a}}_{i,j}^3 = \mathbf{a}_{i,j}^3$ and $\ddot{\mathbf{b}}_{i,j}^3 = \mathbf{b}_{i,j}^3$. This corresponds to the case where the free space monotone curve enters cell i, j through either the corner point i, j , or through the left edge of the cell.
 - Otherwise, we have as in the case $i = 0, j > 0$ above (the monotone increasing curve enters the cell at the bottom edge):
 - * If $\ddot{\mathbf{a}}_{i,j}^0 \geq \mathbf{b}_{i,j}^2$, then $\ddot{\mathbf{a}}_{i,j}^2 = \ddot{\mathbf{b}}_{i,j}^2 = \perp$ as in this case a strict monotone curve cannot reach the top boundary of the cell i, j .
 - * Otherwise if $\ddot{\mathbf{a}}_{i,j}^0 = \mathbf{a}_{i,j}^2$ (and $\ddot{\mathbf{a}}_{i,j}^0 < \mathbf{b}_{i,j}^2$), then $\ddot{\mathbf{a}}_{i,j}^2 = \mathbf{a}_{i,j}^2 + \langle +, 0 \rangle$ (we add a $+$ as the point $\mathbf{a}_{i,j}^2$ itself is unreachable by a strict monotone curve); and $\ddot{\mathbf{b}}_{i,j}^2 = \mathbf{b}_{i,j}^2$.
 - * Otherwise if $\ddot{\mathbf{a}}_{i,j}^0 \neq \mathbf{a}_{i,j}^2$ (and the previous two cases do not hold), then $\ddot{\mathbf{a}}_{i,j}^2 = \max(\ddot{\mathbf{a}}_{i,j}^0, \mathbf{a}_{i,j}^2)$; and $\ddot{\mathbf{b}}_{i,j}^2 = \mathbf{b}_{i,j}^2$.

Note that in this case a monotone curve can enter the free space of cell i, j only from the cell $i, j - 1$.